

**VŠB – Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra informatiky**

**Nástroj pro hromadné testování, odevzdávání projektů a
repositář studijních materiálů**

Tools for Global Testing, Project and Study Materials Repository

Zadání bakalářské práce

Student: **Marek Berger**

Studijní program: B2647 Informační a komunikační technologie

Studijní obor: 2612R025 Informatika a výpočetní technika

Téma: **Nástroj pro hromadné testování, odevzdávání projektů a repositář studijních materiálů**
Tools for Global Testing, Project and Study Materials Repository

Zásady pro vypracování:

Cílem práce je vytvořit webový nástroj pro hromadné testování, odevzdávání projektů a repositář studijních materiálů.

1. Seznamte se s problematikou hromadného testování realizovaného v rámci Cisco Academy, Oracle Academy atd. Analyzujte a navrhnete vhodnou strukturu a funkce pro management otázek, generování testů a jejich automatické vyhodnocení.
2. Seznamte se s problematikou odevzdávání projektů, tedy: definování harmonogramu, podle studijních skupin, proces odevzdávání a schvalování projektu, diskuse nad jednotlivými částmi projektu s vyučujícím a hodnocení projektu.
3. V poslední řadě se zaměřte na repositář studijních materiálů přidružený k jednotlivým kurzům.
4. Součástí řešení bude i možnost sestavení plánu studia, který bude propojovat jednotlivé aktivity pro konkrétní studenty.
5. Implementaci proveďte ve vhodně zvoleném prostředí v návaznosti na přihlašovací a technické možnosti univerzity.
6. Výsledné řešení otestujte v praxi, navrhnete možnosti budoucího rozšíření.

Seznam doporučené odborné literatury:

Nastudujte existující LMS Moodle, Barborka, eLogika.
Dále dle pokynů vedoucího bakalářské práce.

Formální náležitosti a rozsah bakalářské práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

Vedoucí bakalářské práce: **Ing. Radoslav Fasuga, Ph.D.**

Datum zadání: 16.11.2012

Datum odevzdání: 07.05.2013



doc. Dr. Ing. Eduard Sojka
vedoucí katedry

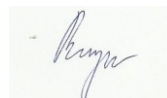


prof. RNDr. Václav Snášel, CSc.
děkan fakulty

Prohlášení studenta

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

Dne: 6. 5. 2013



.....
podpis studenta

Abstrakt

Tématem této bakalářské práce je problematika e – learningu a informačních systému pro zkvalitnění výuky. Cílem práce je navrhnout, implementovat a otestovat vlastní řešení obsahující funkčnost pro generování a organizování testů, odevzdávání a ohodnocení projektů a repositáře studijních materiálů. První část práce je věnována teoretickému základu pro vlastní implementaci. Jsou rozebrány pojmy, které s e – learningem souvisí a popsány stávající informační systémy. Samostatná kapitola je zde i pro shrnutí webových technologií, které poskytuje společnost Microsoft, a které byly použity k implementaci. Ve druhé části práce jsou kapitoly strukturovány dle etap vývoje výsledné aplikace, počínaje specifikací požadovaných funkcí přes analýzu a návrh až k implementaci a testování.

Klíčová slova

E – learning, systém pro řízení průběhu studia, systém pro správu obsahu, online hromadné testování, ASP.NET MVC 3

Abstract

The topic of this bachelor thesis is the issue of e - learning and information systems to improve the quality of teaching. The aim is to design, implement and test a custom solution containing functionality for generating and organization exams, submission and evaluation of projects and repository of learning materials. The first part is devoted to the theoretical basis for its own implementation. They discussed terms which are related to e – learning and described existing information systems. A separate chapter is here for a summary of web technologies that Microsoft provides, and which have been used for implementation. In the second part the chapters are structured according to the phases of development of the resulting application, starting with the specifications required functions through analysis and design to implementation and testing.

Key words

E – learning, learning management system, learning content management system, online global testing, ASP.NET MVC 3

Seznam použitých zkratk

Zkratka	Anglický význam	Český význam
ASP.NET	Active server pages .NET	Technologie pro tvorbu dynamických webových stránek
CRUD	Create, read, update, delete	Vkládání, čtení, úprava, mazání záznamů
EF	Entity framework	Framework zajišťující objektově relační mapování
GPL	General public license	Všeobecná veřejná licence
GUI	Graphical user interface	Uživatelské grafické rozhraní
LCMS	Learning Content Management System	Systém pro tvorbu studijního obsahu
LDAP	Lightweight Directory Access Protocol	Protokol pro přístup k datům na adresářovém serveru
LMS	Learning Management System	Systém pro řízení studia
SQL	Structured Query Language	Strukturovaný dotazovací jazyk
MVC	Model – view – controller	Softwarová architektura
ORM	Object – relational mapping	Objektově relační mapování
PHP	PHP hypertext preprocessor	PHP hypertextový preprocesor
SCORM	Shareable Content Object Reference Model	Model sdílených obsahových objektů
URL	Uniform Resource Locator	Jednoznačný identifikátor zdroje

Obsah

1	Úvod.....	1
2	E – learning.....	2
2.1	Co je to e –learning	2
2.2	Historie e - learningu.....	2
2.3	Virtuální univerzita	3
2.4	System LMS	3
2.5	System LCMS.....	4
2.6	Výhody a nevýhody E - learningu	4
3	Existující LMS systémy	5
3.1	Moodle	5
3.2	iTutor.....	6
3.3	Oracle Academy	6
3.4	Cisco Networking Academy.....	7
3.5	Microsoft Virtual Academy.....	8
4	Webové technologie platformy .NET	9
4.1	Co je to ASP.NET?.....	9
4.2	ASP.NET MVC Framework.....	9
4.2.1	MVC Architektura.....	9
4.2.2	Novinky v ASP.NET MVC 3	10
4.3	ORM.....	10
4.3.1	Entity Framework.....	11
5	Analýza požadavků a jejich specifikace	12
5.1	K čemu má nový systém sloužit	12
5.2	Kdo bude se systémem pracovat.....	12
5.3	Vstupy	12
5.4	Výstupy	13
5.5	Funkce systému	13
5.6	Nefunkční požadavky.....	14
6	Návrh aplikace	15
6.1	Případy užití.....	15
6.2	Diagram případu užití	18
6.3	Proces vyhodnocování testu	19

6.3.1	Procentuální vyhodnocení.....	19
6.3.2	Bodové vyhodnocení.....	20
6.4	Doménový model.....	20
6.5	Lineární zápis typů databázových entit.....	23
7	Implementace.....	25
7.1	Přístup k datům.....	25
7.2	Uchovávání historie předmětů a událostí.....	26
7.3	Zabezpečení.....	27
7.3.1	Autentifikace.....	27
7.3.2	Autorizace.....	28
7.4	Testování.....	28
7.4.1	Generování testů.....	28
7.4.2	Vyhodnocování testů.....	28
7.5	Odevzdávání projektů a nahrávání studijních materiálů.....	30
7.5.1	Adresářová struktura pro ukládání souborů.....	30
7.5.2	Stavy odevzdání.....	31
7.6	Diagram komponent.....	31
8	Aplikace.....	33
8.1	Uživatelské rozhraní.....	33
8.1.1	Navigační menu předmětu.....	33
8.1.2	Navigační menu testu.....	33
8.1.3	Navigační menu projektu.....	35
8.2	Možnosti rozšíření.....	36
8.2.1	Videokonference a možnost sdílení plochy.....	36
8.2.2	Graf zobrazující statistiky úspěšnosti otázek.....	36
8.2.3	Přenositelnost studijních materiálů.....	36
8.2.4	Další vyhodnocovací metody.....	36
8.2.5	Implementace webové služby.....	37
8.3	Testování.....	37
9	Závěr.....	38
	Použitá literatura.....	39
	Seznam obrázků.....	40
	Seznam rovnic.....	40
	Seznam tabulek.....	40

Seznam příloh.....	41
--------------------	----

1 Úvod

Proces vzdělávání lze chápat jako cílevědomý rozvoj technických a vědeckých znalostí, získávání nových praktických zkušeností a učení se, jak tyto znalosti uplatnit v praxi. Trh práce poskytuje nespočet pracovních příležitostí s různým zaměřením a různými požadavky na míru dosaženého vzdělání, což vytváří prostor pro využití systémů a nástrojů, pomocí nichž lze požadovaného stupně dosáhnout.

V posledním desetiletí zaznamenáváme prudký rozvoj v oblasti informačních a komunikačních technologií, který v procesu vzdělávání otevřel nové možnosti. Zejména u formy distanční výuky. S využitím dnešních softwarových a hardwarových možností se již nemusí výuka realizovat pomocí klasického modelu, kdy pedagog přednáší pouze pro studenty v posluchárně, nýbrž může přednášet pro studenty, kteří jsou v libovolné lokalitě po celém světě. Informační technologie ale nachází uplatnění i v prezenční formě ke zkvalitnění výuky, například pomocí mediálních prostředků, které studentům pomohou pochopit látku nebo pomocí nástrojů pro jejich testování a ohodnocení, což nás přivádí k tématu této bakalářské práce.

Podrobnější informace o problematice nástrojů ke zkvalitnění výuky a e – learningu jsou rozepsány ve druhé kapitole. Třetí kapitola je zaměřena na již existující systémy, jejich vlastnosti a funkce. Čtvrtá kapitola shrnuje webové technologie platformy .NET. Od páté kapitoly je pozornost zaměřena na vývoj konkrétní aplikace, jež byla cílem této práce, počínaje specifikací požadovaných funkcí. Návrh aplikace je obsahem šesté kapitoly. Sedmá kapitola je věnována implementaci a kapitola osmá již popisuje výslednou aplikaci, její uživatelské rozhraní, možnosti dalšího rozšíření a výsledky testování.

2 E – learning

V této kapitole je popsána problematika e - learningu a rozebrány pojmy, které s ním souvisí.

2.1 Co je to e –learning

Pojem e - learning nemá jasně vymezenou definici a lze na něj nahlížet z několika uhlů. Je ovšem důležité, že představuje alternativu ke klasickému způsobu vyučování, kdy se předpokládá fyzická přítomnost, jak učitele, tak studentů a jejich vzájemná přímá komunikace. Zde je uvedeno několik definic, jak lze e-learning chápat [1]:

- E - learning je vzdělávací proces, využívající informační a komunikační technologie k tvorbě kursů, k distribuci studijního obsahu, komunikaci mezi studenty a pedagogy a k řízení studia.
- E - learning je forma vzdělávání využívající multimediální prvky - prezentace a texty s odkazy, animované sekvence, video snímky, sdílené pracovní plochy, komunikaci s lektorem a spolužáky, testy.
- E - learning je v podstatě jakékoli využívání elektronických materiálních a didaktických prostředků k efektivnímu dosažení vzdělávacího cíle s tím, že je realizován zejména/nejenom prostřednictvím počítačových sítí.

E – learning získává tedy nejširší uplatnění u distanční formy studia, ovšem i u klasické prezenční formy lze e – learning použít ke zkvalitnění výuky. Zde dochází k častému zaměňování pojmu e – learning s pojmem „online výuka“, která předpokládá pouze online spojení mezi pedagogem a studenty.

2.2 Historie e - learningu

V druhé polovině šedesátých let se začalo experimentovat se stroji na učení, byl to počátek tzv. vyučovacích automatů. U nás byl vyvinut jeden z nejlepších a nesl název Unitutor. Unitutor obsahoval několik stránek, kde každá stránka byla výkladem učiva, na jejímž konci se nacházela kontrolní otázka s výběrem z několika možných odpovědí. Stránky, bylo možné dále větvit podle provedené volby odpovědi. Automaty byly ovšem příliš složité a ne příliš účinné, proto se příliš neujaly.

Ve druhé polovině osmdesátých let se začínají objevovat první šestnáctibitové počítače. Dochází ke značnému rozvoji aplikací, převážně kancelářských, a počítače začínají být dostupné i pro domácnosti. Současně se objevují i programy, které navazují na Unitutor. Takové programy většinou začínaly přihlášením tj. zapsáním jména zkoušeného. Následoval sled za sebou jdoucích otázek, na které se odpovídá výběrem z nabídnutých možností. Odezva počítače informuje o správnosti volby, otázka je nějakým způsobem ohodnocena a pak se přejde na další. Lepší programy tohoto typu dovedou pořadí otázek i nabízených odpovědí různě náhodně pomíchat nebo je náhodně vybírat z většího počtu připravených. Pro učitele se někdy ukládá taky informace o úspěšnosti u jednotlivých otázek [2].

Využití počítačů pouze ke zkoušecímu procesu ovšem nestačilo a bylo značně kritizováno. Bylo třeba nějakým způsobem řídit a kontrolovat, průběh studia jednotlivých studentů. Proto se

v devadesátých letech začaly takové aplikace obohacovat o prvky umělé inteligence. Za vývojem stály převážně univerzitní vědecké týmy. Aplikace v sobě spojovaly jak výklad učiva, tak procvičování látky a testování. Všechny tyto prvky bylo možné vzájemně propojovat a vytvářet tak samostatné kurzy. Navíc technologie umožňovala doplnit takové aplikace o grafické prvky, animace, interaktivní aplikace, zvukové efekty apod. Výsledky studentů se uchovávaly a vyhodnocovaly.

Vývoj na univerzitách šel rychle dopředu. Studijní materiály, obsahy přednášek a jiné zdroje byly publikovány na internetu nebo na místních sítích univerzit. Na přelomu dvacátého a jednadvacátého století vznikají virtuální univerzity. Plně zaměstnaní dospělí tedy mohli studovat svým vlastním tempem a jen s pomocí svého počítače a přístupu k internetu.

2.3 Virtuální univerzita

Pojem virtuální univerzita představuje organizaci, která prostřednictvím internetu a informačních systémů umožňuje získat plnohodnotné vysokoškolské vzdělání. Odlišnost od klasického studia je zřejmá, a sice že, studenti fyzicky nenavštěvují prostředí univerzity, nýbrž ke studiu jim postačí připojení k internetu. Stejně jako ostatní studenti, tak i studenti virtuální univerzity absolvují cvičení, dodržují harmonogram a jsou povinni vypracovávat testy apod.

Virtuální univerzity představují příklady, jak využít informační technologie v oblasti vzdělávání a to jak v rozvinutých, tak v rozvojových zemích. Mají globální charakter a nabízí se studentům i mimo oblast či zemi své působnosti. Další výhodou může být snadná a bohatá možnost spolupráce mezi univerzitami a to i na mezikontinentální úrovni.

S virtuální univerzitou je ovšem spojeno i několik problémů. V první řadě jde o problémy samotného internetu (zabezpečení, ochrana autorských práv a duševního vlastnictví, dostatečná konektivita, apod.) [3].

2.4 Systém LMS

LMS je zkratkou anglických slov „Learning Management System“ což v doslovném překladu znamená „učební správní systém“. Podstata LMS je organizovat a řídit výuku a kompetence. Z pohledu studenta je LMS virtuální prostředí, kde nalezne pokyny a informace ke studiu a může zde být nějakým způsobem ohodnocen, nejčastěji vykonáváním testů, odevzdáváním projektů, případně vypracováním různých cvičných úkolů apod.

Funkcionalita LMS systému se může v závislosti na výrobci lišit, avšak od kvalitního LMS se většinou očekává následující [4]:

- Řízení a evidenci všech typů výuky, od elektronických asynchronních kurzů, přes virtuální učebny až po klasickou výuku v učebnách.
- Centrální katalog všech vzdělávacích akcí (elektronické kurzy, virtuální třídy/videokonference, učebny), externí výuka, správa zdrojů a financí s tím spojenou.
- Modelování organizace a kompetencí, evidování dosažených individuálních dovedností

- Zpřístupňování vzdělávacích akcí, sledování aktivit jednotlivých uživatelů od souhrnu po detaily, reportování všech typů výukových aktivit společně i jednotlivě.
- Bohatou sadu synchronních a asynchronních komunikačních kanálů mezi studenty, lektory a manažery studia, prostředky pro zachytávání výměny a sdílení informací a znalostí

Souhrn některých existujících LMS systémů je obsažen ve třetí kapitole.

2.5 Systém LCMS

Termín LMCS z anglického „Learning Content Management System“ označuje systém pro správu a tvorbu výukových materiálů. Na rozdíl od LMS nezasahuje do řízení výuky jako takové, stará se pouze o její obsah. LMS a LCMS lze tedy do jisté míry považovat za vzájemně nezávislé systémy.

Oba systémy by měly podléhat normě SCORM (Shareable Content Object Reference Model), která definuje popis a chování obsahu tak, aby byl přenositelný mezi různými LMS systémy. Z pohledu normy SCORM je obsah definován jako „vzdělávací objekt“ což je objekt, který je schopen opětovně sdělovat nějaké výukové informace [4].

2.6 Výhody a nevýhody E - learningu

Zavedením e – learningu ve vzdělávací instituci má za následek snížení nákladů na klasickou formu vzdělávání. Především jde o náklady na provoz učeben, zajištění studijních materiálů, mzdy pedagogů, dopravy a další. Přínosem na straně studentů je flexibilita a časově nezávislé individuální studium, kdy si student sám volí, kdy a jak dlouho se látce bude věnovat a absolvuje kurzy dle vlastních potřeb.

Výhodou může být i neosobitý přístup mezi studenty a pedagogy, hodnocení studentů je potom spravedlivé a dle stejných pravidel, nehledě na to, že testy jsou vyhodnocovány automaticky.

Náklady na zavedení e – learningu mohou být ovšem vysoké a odrazující. Je třeba zajistit potřebné hardwarové vybavení, zavést vhodné LMS systémy, zajistit výukové materiály a podobně.

3 Existující LMS systémy

Obsahem této kapitoly je stručných přehled některých existujících LMS systému. Shrnuty jsou jejich vlastnosti, případně výhody a nevýhody.

3.1 Moodle

Moodle je softwarový balík určený pro podporu prezenční i distanční výuky prostřednictvím online kurzů dostupných na internetu. Pro realizaci výuky poskytuje možnost publikovat studijní materiály, zavádět diskusní fóra pro komunikaci mezi účastníky a pedagogy, hodnocení a odevzdávání elektronicky sepsaných dokumentů, tvorbu online testů a další podobné nástroje.

Jako původní autor Moodle je označován Martin Dougiamas, který se podílí na vývoji i dnes. První prototyp byl napsán v jazyce Python. Aktuální verze Moodle (2.41) je implementována pomocí PHP verze 5.3.2, pro přístup k datům je možno použít databáze MySQL, Postgres, MSSQL, Oracle. Moodle je dostupný zdarma jako open source software spadající pod obecnou veřejnou licenci GPL [5].

Na stránkách <http://demo.moodle.net/> je dostupné demo aplikace k vyzkoušení pro kohokoliv. Je zde připraveno několik uživatelských rolí, ke kterým se lze v rámci zkušební verze přihlásit (admin, manager, teacher, student).

The screenshot shows the Moodle interface for adding a multiple choice question. The sidebar on the left contains navigation links (Home, Site pages, My profile, Current course, Z, Participants, Reports, General, 7 April - 13 April, Test1, Info, Results, 14 April - 20 April, 21 April - 27 April, 28 April - 4 May, 5 May - 11 May, 12 May - 18 May, 19 May - 25 May, 26 May - 1 June, 2 June - 8 June, 9 June - 15 June, Courses) and settings (Quiz administration, Edit settings, Group overrides, User overrides, Edit quiz, Preview, Locally assigned roles, Permissions, Check permissions, Filters, Logs, Backup, Restore, Question bank, Course administration, Switch role to..., My profile settings, Site administration). The main content area is titled 'Adding a Multiple choice question' and contains the following fields:

- Category: Default for Z
- Question name*: Hora
- Question text: Jaká je nejvyšší hora České republiky?
- Path: p
- Default mark*: 1
- General feedback: (empty text area)
- Path: p
- One or multiple answers?: Multiple answers allowed
- Shuffle the choices?: ☒
- Number the choices?: a, b, c, ...
- Choice 1:
 - Answer: (empty text area)
 - Path: p
 - Grade: None
 - Feedback: (empty text area)
 - Path: p

Obrázek: 3.1 – LMS Moodle – tvorba otázky

Zdroj: <http://demo.moodle.net/>

Uživatelské rozhraní demo aplikace je značně nepřehledné, a strávil jsem poměrně dlouhou dobu hledáním a procházením, než jsem pochopil provádění základních operací. Moodle ovšem poskytuje rozsáhlou wiki nápovědu.

3.2 iTutor

iTutor je platforma pro vzdělávání, sdílení vědomostí a spolupráci. Jedná se o řešení s centrální databází a s jednotnou bezpečnostní infrastrukturou umožňující efektivně organizovat a řídit celý vzdělávací proces, zahrnující samostudium elektronických kurzů a materiálů, synchronní vzdělávání ve virtuálních třídách a klasickou výuku na učebnách. Modulární architektura umožňuje postupně integrovat jednotlivé funkční bloky, dle požadavků [6].

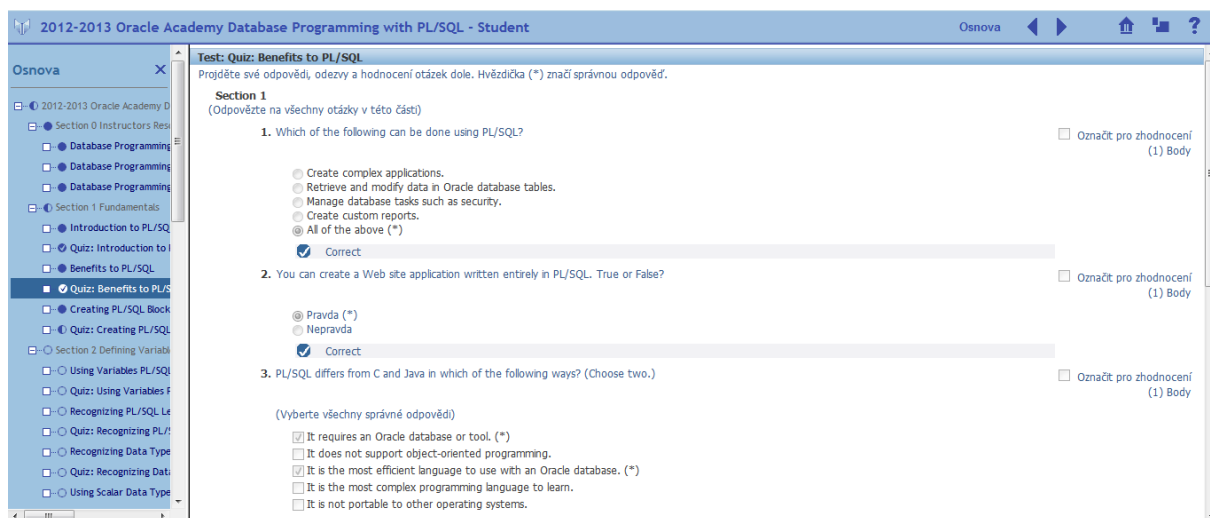
The screenshot shows the iTutor tester interface. On the left is a sidebar titled 'Témata a otázky' with tabs for 'Otázky', 'Předpisy', 'Testy', and 'Hledání'. It contains a tree view of topics including 'Základní téma', 'Bezpečnost práce', 'Datart', 'GE', 'Kompetence', 'Makro - demo', 'McDonalds', 'Ukázka ČSA', 'VB úvěry', 'Zeměpis', 'Čechy', 'Hory', 'Řeky', and 'Morava'. The main area is titled 'Otázka:' and has tabs for 'Vlastnosti', 'Volby', 'Poznámka', 'Multimedia', 'Zpožděné odpovědi', and 'Teorie'. It shows a question creation form with three text questions. Each question has a 'Text' field, a 'Počet číselných odpovědí' dropdown (set to 3), and two 'Odpověď' fields with 'Interval od: a do: včetně' labels. The 'Skóre' field is set to 1 for correct answers and -1 for incorrect ones. A 'Uložit' button is at the bottom right. The footer indicates 'Optimalizováno pro rozlišení 1024x768, MS Windows, Internet Explorer. Copyright 2000-2010, Kontis s.r.o.'

Obrázek: 3.2 – iTutor tester – tvorba otázky

Zdroj: <http://www.kontis.cz/img/moduly/tester.gif>

3.3 Oracle Academy

Oracle Corporation je společnost vyvíjející relační databáze působící na trhu již od roku 1977 [7]. Společnost provozuje e – learningový portál poskytující kurzy nejen z oblasti relačních databází ale i z oblasti technologie JAVA. Úspěšným dokončením kurzů získá absolvent certifikaci. Kurzy jsou členěny do jednotlivých kapitol, obsahující materiály ve formě prezentací, případně krátké kvízy s definovaným minimálním procentuálním ziskem (Obrázek: 3.3). Kvízy je možno opakovat několikrát. Kurz je zakončen finálním testem (Final exam).

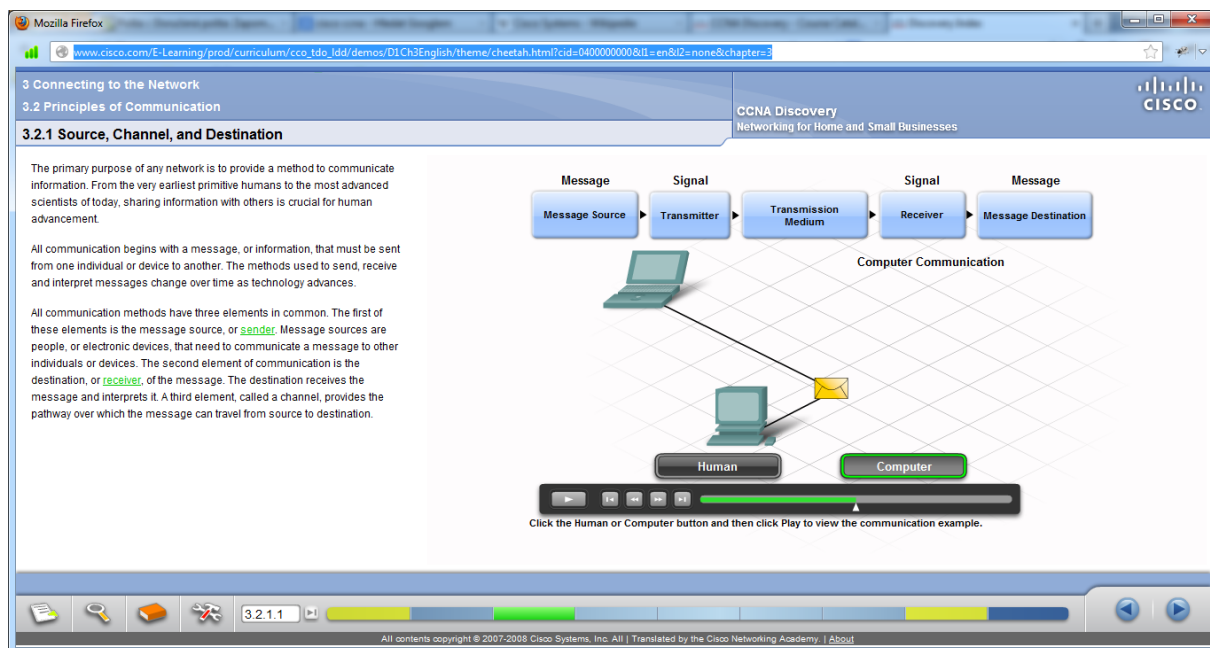


Obrázek: 3.3 – Oracle Academy – vypracování testu

Zdroj: <http://ilearning.oracle.com/>

3.4 Cisco Networking Academy

Cisco Systems, Inc je americká mezinárodní společnost sídlící v San Jose v Kalifornii, zabývající se návrhem, výrobou a prodejem síťového vybavení. Poskytuje možnost získat v oblasti počítačových sítí řadu certifikátů různých úrovní a zaměření. Výukové kurzy obsahují velmi kvalitní výukové materiály. Jde o interaktivní prezentace vytvořené pomocí technologie Adobe Flash, doplněné o videa, animace a práci s externími programy.



Obrázek: 3.4 – Cisco Networking Academy - studijní materiály

Zdroj: <http://www.cisco.com>

3.5 Microsoft Virtual Academy

Nabídka certifikací společnosti Microsoft je velice pestrá. Společnost vytvořila webovou aplikaci, jež má za cíl připravit studenty na certifikační testy. Aplikace obsahuje několik kurzů s různým zaměřením od práce s Microsoft Office přes správu Windows serverů až po vývoj aplikací pro různé platformy a je dostupná na adrese <http://microsoftvirtualacademy.com>. Výukové materiály jsou členěny do jednotlivých kapitol a publikovány formou krátkých prezentací a videí (Obrázek: 3.5). Kurz může obsahovat sadu testů ověřující znalosti. Za úspěšné dokončení kurzů jsou připisovány fiktivní body, které určují pozici v porovnání s ostatními uživateli, což dle mého názoru, může být velice motivující. V současné době je registrováno přes milion uživatelů.

The screenshot displays the Microsoft Virtual Academy interface. At the top, there's a navigation bar with links like 'Microsoft Virtuální Akademie', 'Kurzy', 'Informační panel', 'Nejlepší studenti', and 'Začínáme'. The main content area features the course 'Building Web Apps with ASP.NET Jump Start' with a user rating of 5 stars (25 reviews) and a completion progress of 32%. Below this, a list of modules is shown, with the first three highlighted: '01 | What's New in ASP.NET 4.5', '02 | Building and Deploying Websites with ASP.NET MVC 4', and '03 | Creating HTML5 Applications with jQuery'. To the right, there's a section for 'Obľíbené kurzy' (Favorite courses) and 'Nejlepší studenti' (Top students). At the bottom, a 'Celková aktivita' (Overall activity) section shows statistics: 1,129,098 registered students, 1,619,598 completed tests, and 1,642,411 training hours.

Obrázek: 3.5 – Microsoft Virtual Academy

Zdroj: <http://microsoftvirtualacademy.com>

4 Webové technologie platformy .NET

Kapitola shrnuje technologie, které jsou v rámci platformy .NET k dispozici pro tvorbu webu, a které byly použity v rámci této práce.

4.1 Co je to ASP.NET?

ASP.NET nebo Active Server Pages .NET je součástí platformy .NET Framework. Jedná se o technologii, která umožňuje dynamicky vytvářet a spravovat dokumenty na webovém serveru. Převážně se jedná o HTML dokumenty.

Oproti starší technologii ASP se kód neinterpretuje, nýbrž kompiluje, což ve výsledku představuje vyšší výkon aplikace a také snadnější odhalení chyb už při vývoji. ASP.NET také zajišťuje plnohodnotný objektový přístup a jednoduché použití knihoven .NET Frameworku. První verzi Microsoft uvolnil v roce 2002 označenou jako ASP.NET Web Forms 1.0, ovšem nesla sebou několik úskalí za, které byla značně kritizována [8].

- **Velikost view state:**
Pokud se v aplikaci očekává interakce od uživatele, je třeba si nějakým způsobem ukládat vstupní informace, pomocí kterých je pak uživateli předán požadovaný výsledek. Jelikož je ale http protokol bezstavový, ASP.NET řeší tento problém odesíláním skrytého pole, zvaného view state, při každém požadavku. Toto pole může obsahovat až stovky kilobajtů.
- **Limitovaná kontrola html výsledku:**
ASP.NET serverové prvky jsou ve výsledku generovány automaticky, a výsledek vždy nemusí být dle očekávání.

4.2 ASP.NET MVC Framework

V říjnu 2007 byl na konferenci v Austinu v Texasu poprvé představen ASP.NET MVC framework, jako přímá odpověď na rozvoj webových technologií (např. Ruby on rails) a kritiku Web Forms [8].

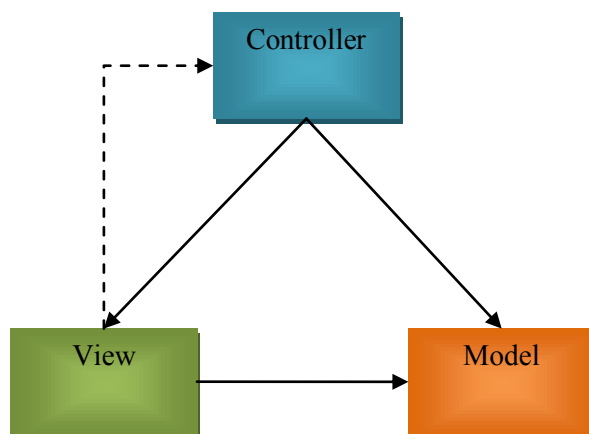
4.2.1 MVC Architektura

MVC architektura nebyla zavedena společně s ASP.NET MVC, její první představení se datuje na rok 1970. Základní myšlenkou je oddělení uživatelského grafického rozhraní GUI a logiku aplikace. Architektura je definována pomocí tří oddělených částí:

- **Model:**
Reprezentuje nebo přímo uchovává samotná data. Z pohledu objektové implementace je to třída, která definuje atributy, reprezentující data a metody pro práci s nimi.
- **View:**
Grafická podoba výstupu aplikace. Je to tedy nástroj, se kterým uživatel nějakým způsobem pracuje, aby dostal požadovaný výsledek. Konkrétně u webových technologií typicky obsahuje Html kód, který je po zpracování MVC aplikací interpretován uživateli

- Controller:
Směřuje uživatelské požadavky, zajišťuje požadovaná data a přiřazuje k nim patřičný výstup (View).

Vzájemnou komunikaci jednotlivých komponent MVC architektury znázorňuje Obrázek: 4.1.



Obrázek: 4.1 – MVC architektura

4.2.2 Novinky v ASP.NET MVC 3

Za hlavní novinku MVC 3 se dá považovat integrace Razor View Engine, který přináší větší komfort v použití view šablon.

V předchozích verzích byly dynamické bloky ohraničené pomocí znaků `<%` a `%>`, v Razor View engine bylo toto nahrazeno znakem `@`. Tato notace poskytuje jednodušší použití, flexibilnější funkčnost (Razor helpers) a rychlejší kompilaci.

V poslední době je velmi populární použití tzv. dependency injection. Jde o návrhový vzor, který výrazně redukuje závislost mezi konkrétními objekty. ASP.NET MVC 3 obsahuje zvýšenou podporu pro použití dependency injection (Dependency resolver) [8].

4.3 ORM

Objektově relační mapování neboli ORM je technika, která zajišťuje konverzi mezi relační databází a objektově orientovaným programovacím jazykem. Většina ORM frameworků zajišťuje selekci dat jen pomocí objektových metod, což může zapříčinit jistou nezávislost na použité databázi. Použití ORM zjednodušuje provádění základních CRUD operací (create, read, update, delete) a usnadňuje práci s daty. Nevýhodou ORM je vyšší výkonová náročnost, jelikož je třeba pro každou entitu vytvořit samostatný objekt.

4.3.1 Entity Framework

Entity framework (dále jen EF) je jeden z nástrojů, které řeší objektově relační mapování. Je určen pro platformu .NET a aktuálně je uvolněna verze 5.0.0.

EF podporuje techniku code first, kdy framework přebírá zodpovědnost za vytvoření a manipulaci s databází. Vývojář tak pouze vytvoří doménový model požadované aplikace, a databáze bude vygenerována zcela automaticky podle příslušných tříd a definovaných pravidel. EF s prvním vytvořením databáze vytvoří i meta tabulku, kde se ukládá stav aktuálně používaného doménového modelu. Pokud se tedy stav změní, EF to porovnáním hodnot s meta tabulkou rozpozná a je vyslán příkaz na znovuvytvoření databáze. Toto chování je ovšem u již uvolněných aplikací nepřijatelné, ale při vývoji značně ulehčuje programátorovi práci.

Pro získávání dat používá EF návrhový vzor Identity map, který zajišťuje načtení konkrétní entity pouze jednou.

5 Analýza požadavků a jejich specifikace

5.1 K čemu má nový systém sloužit

Systém spadá do definice LMS nástrojů a umí tedy využívat prostředky e – learningu ke zkvalitnění výuky jak prezenční, tak distanční formy studia. Funkčnost systému zahrnuje kontrolu a řízení průběhu studia pro konkrétní studenty, svázané do jednotlivých kurzů / předmětů. Student může být bodově ohodnocen za vykonávání testů a za odevzdávání projektů. Jeho dosažené výsledky se do systému evidují a jsou uspořádány podle akademického roku.

Součástí řešení je i administrační rozhraní pro oprávněné osoby. Umožňuje správu testových otázek a správu generovaných testů, přičemž je možné pro otázky vytvářet určitá omezující pravidla, která jsou při generování zohledněna. Dále umožňuje definici komplexních projektů skládajících se z jednotlivých úkolů, kdy každý úkol lze v rámci rozhraní konzultovat, hodnotit a obodovat.

5.2 Kdo bude se systémem pracovat

Systém předpokládá víceuživatelský přístup. Uživatele je možno rozdělit na přihlášené a anonymní. Pro účely systému to ovšem nestačí a je nutné pro přihlášené uživatele jasně definovat oprávnění a privilegia pomocí uživatelských rolí.

Systém definuje tyto role:

- Student:
Má právo odevzdávat a konzultovat jednotlivé části projektů s instruktory. Má právo vykonávat test v předem zadaných termínech.
- Cvičící:
Má právo hodnotit, schvalovat a konzultovat projekty se studenty na základě přiřazení do studijní skupiny. Má právo ve svých skupinách spravovat události (projekty, testy, aktuality, studijní materiály).
- Administrátor:
Má právo spravovat skupiny, definovat přiřazení uživatelů do skupin a definovat role uživatelů. Spravuje předměty, události, otázkové sety testů. Má právo hodnotit, schvalovat a konzultovat projekty se studenty.

5.3 Vstupy

- Předmět – Zkratka, Název, popis, rok zavedení
- Kategorie – Krátký název, dlouhý název, rodičovská kategorie
- Skupina – Název skupiny, akademický rok, klíč k zápisu do skupiny
- Aktualita – Titulek, text, rok zavedení, rok ukončení, přístupné skupiny
- Test – Název, popis, minimální počet bodů, maximální počet bodů, aktivní šablona, rok zavedení, rok ukončení, přístupné skupiny

- Šablona – Název, metoda vyhodnocení, mód přepočtu bodů, mód zaokrouhlování, indikace zaokrouhlování po jednotlivých otázkách, indikace záporného hodnocení, časový limit, otázkové sety (množství otázek, počet bodů za otázku, indikace přiřazených a vyřazených otázek)
- Termín testu – Datum zahájení, datum ukončení, počet pokusů, uživatel
- Otázkový set – název
- Otázka – Text, typ otázky, indikace zda lze k otázce přidat komentář
- Odpověď – Text odpovědi, indikace zda je odpověď pravdivá
- Přístupná síť – IP adresa sítě, maska v CIDR formátu
- Projekt – Název, popis, minimální počet bodů, maximální počet bodů, rok zavedení, rok ukončení, přístupné skupiny
- Úkol projektu – Název, popis, minimální počet bodů, maximální počet bodů, formáty
- Termín odevzdání projektu – Datum zahájení, datum ukončení, uživatel
- Odevzdaný úkol – Stav, získané body, soubor
- Zpráva – Text, indikace zda je zpráva veřejná, příjemce zprávy
- Studijní materiál – Název, popis, soubor

5.4 Výstupy

- Výpis předmětů pomocí stromové struktury kategorií
- Přehled událostí (aktivita, test, projekt, studijní materiál) pro jednotlivé předměty
- Prostředí pro testování studentů pomocí automaticky generovaných testů
- Přehled výsledků testů detailně pro konkrétní studenty i souhrnně pro všechny
- Prostředí pro odevzdání úkolu
- Přehled odevzdaných souborů k danému projektu
- Prostředí pro konzultaci konkrétních úkolů mezi studentem a pedagogy

5.5 Funkce systému

- Správa předmětů
- Správa projektů
 - Správa úkolů
 - Správa termínů na odevzdání úkolu
- Hodnocení projektu
 - Změna stavu projektu
 - Zasílání zpráv k danému úkolu

- Odevzdání verze projektu
- Správa testů
 - Správa testových šablon
 - Správa testových otázek
 - Správa odpovědí
 - Přiřazení otázek k šabloně
 - Volba aktivní šablony
 - Správa termínů k vypracování testu
 - Uzamykání testů
- Vykonání testu
 - Ukládání zvolených odpovědí
 - Vyhodnocení testů podle zadaných pravidel
- Správa studijních materiálů
- Správa aktualit
- Správa uživatelů
- Správa studijních skupin
 - Automatický zápis studentů do skupin
 - Přesun uživatelů do skupin

5.6 Nefunkční požadavky

Systém je realizován jako webová aplikace s vazbou na univerzitní systém pomocí LDAP (Lightweight Directory Access Protocol) protokolu [9].

6 Návrh aplikace

6.1 Případy užití

Případy užití popisují určitou funkci jako seznam kroků, který definuje interakci mezi systémem a uživatelem neboli aktérem. Každý případ užití zde má vyčleněny podmínky, za kterých může být vykonán (prekondice). Některé administrační funkce jsou vynechány a budou popsány až v pozdějších kapitolách, případně jde o základní CRUD operace a jejich detailní popis není předmětem této práce. Všechny případy užití graficky znázorňuje diagram případu užití (Obrázek: 6.1).

6.1.1 Automatický zápis do skupiny

Aktéři: Student

Prekondice:

- Aktér je přihlášen do systému
- 1. Aktér zvolí předmět
- 2. Systém ověří, zda již není na předmětu přihlášen (je zapsán do jedné z náležících skupin)
- 3. Aktér vyplní klíč k zápisu a odešle formulář
- 4. Systém prohledá skupiny k danému předmětu v aktuálním akademickém roce
 - a. Pokud je nalezena skupina se shodným klíčem, je přidán záznam o náležitosti aktéra do této skupiny

6.1.2 Odevzdání projektu

Aktéři: Student

Prekondice:

- Aktér je přihlášen do skupiny v aktuálním akademickém roce
- Aktér ještě neodevzdal žádný soubor, nebo jeho projekt ještě není uzavřen
- Aktuálně je dostupný termín pro daného aktéra
- 1. Aktér zvolí úkol, vybere požadovaný soubor a odešle webový formulář
- 2. Systém zkontroluje typ odevzdaného souboru
- 3. Systém uloží záznam do databáze, odeslaný soubor je uložen do adresářového systému
 - a. Pokud je stav nastaven na „K dopracování“, je uložena nová verze
 - b. Pokud je stav nastaven na „Odevzdaný“, je přepsán poslední odevzdaný soubor
 - c. Pokud pro aktéra neexistuje záznam, je vytvořen nový

6.1.3 Konzultace k úkolu projektu

Aktéři: Student, instruktor, administrátor

Prekondice:

- Aktér je přihlášen do skupiny (Student, instruktor)

1. Aktér přejde na stránku s konverzací s konkrétním uživatelem
2. Systém vyhledá zprávy pro konkrétní úkol, kde je aktér určený jako příjemce nebo jako odesílatel
 - a. Všechny zprávy ve výsledku označené jako nepřečtené jsou upraveny a označeny jako přečtené

6.1.4 Napsání zprávy

Aktéři: Student, instruktor, administrátor

Prekondice:

- Aktér je přihlášen do skupiny (Student, instruktor)
1. Aktér napíše zprávu a odešle webový formulář
 - a. Aktér může zvolit, zda má být daná zpráva veřejná (instruktor, administrátor)
 2. Systém uloží zprávu do databáze a přidá k příslušným příjemcům
 - a. Pokud je zpráva neveřejná, je přidán pouze konkrétní příjemce
 - b. Pokud je zpráva veřejná, jsou jako příjemci označeni všichni uživatelé, kteří jsou aktuálně přihlášení do příslušných skupin

6.1.5 Vykonání testu

Aktéři: Student

Prekondice:

- Aktér je přihlášen do skupiny v aktuálním akademickém roce
 - Aktuálně je dostupný pokus pro daného aktéra
 - K testu je přiřazena šablona
1. Aktér si vyžádá vygenerování testu
 2. Systém podle aktivní šablony vygeneruje otázky
 - a. Systém vyhodnotí přiřazené a odřazené otázky
 - b. Systém projde všechny přiřazené sety a náhodně vygeneruje příslušný počet otázek za daný počet bodů
 3. Systém uloží záznam o novém pokusu do databáze a zobrazí test

6.1.6 Změna odpovědi

Aktéři: Student

Prekondice:

- Daný test není uzavřen
1. Aktér zvolí odpověď
 2. Systém zaznamená asynchronní požadavek a uloží odpověď do databáze
 - a. Kontrola zda nevypršel čas

6.1.7 Přidání komentáře k otázce

Aktéři: Student

Prekondice:

- Daný test není uzavřen
 - Otázka lze komentovat (je nastaven příslušný atribut)
1. Aktér napíše komentář
 2. Systém při opuštění textového pole komentáře zaznamená asynchronní požadavek a uloží komentář do databáze

6.1.8 Hodnocení projektu

Aktéři: Instruktor, Administrátor

Prekondice:

- Aktér je přihlášen do příslušné skupiny (Instruktor)
 - Student odevzdal alespoň jednu verzi úkolu
1. Systém zobrazí všechny odevzdané verze úkolu daného studenta
 2. Aktér vybere stav, případně zadá konečný počet bodů a napíše zhodnocení
 3. Systém uloží hodnoty do databáze
 - a. Kontrola, zda jsou body ve správném rozsahu s ohledem na zvolený stav

6.1.9 Uzamykání / odemykání testu

Aktéři: Instruktor, Administrátor

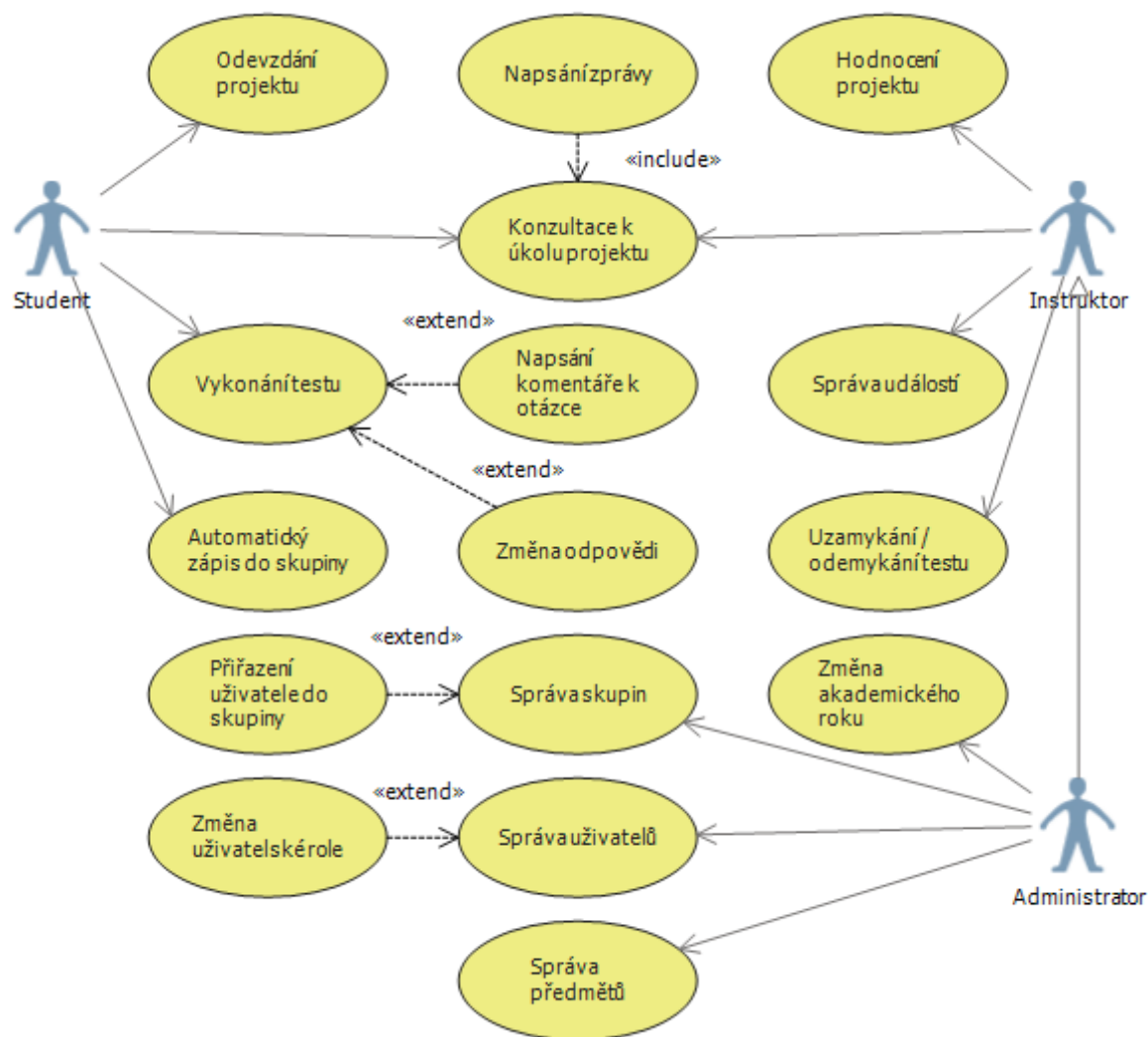
Prekondice:

- Aktér je přihlášen do příslušné skupiny (Instruktor)
 - Test je spuštěný / uzamčený
1. Aktér zvolí konkrétního studenta, u kterého chce test uzamknout / odemknout
 2. Systém upraví stav pokusu na uzamčený¹ / běžící

¹ Student rozpozná změnu, až když se pokusí komunikovat se serverem.

6.2 Diagram případu užití

Diagram případu užití graficky znázorňuje vazby mezi aktéry systému a funkcemi, jež jsou oprávněni vykonávat. Vazba `<<include>>` znázorňuje vložení jednoho případu užití do druhého, to znamená, že pokud se provede první případ užití, provede se automaticky i druhý. Vazba `<<extend>>` znázorňuje rozšíření případu užití, to znamená, že pokud se provede rozšířený případ užití, může se nepovinně provést i ten rozšiřující.



Obrázek: 6.1 – Diagram případu užití

6.3 Proces vyhodnocování testu

Každý test je v první řadě hodnocen procentuálně, a to součtem procentuální úspěšnosti u jednotlivých otázek. Následně se podle maximálního počtu bodů vypočítá bodový zisk jak za jednotlivé otázky, tak celkově za celý test. Vyhodnocení testu je možno ovlivnit několika parametry, které jsou nastaveny v aktivní šabloně.

6.3.1 Procentuální vyhodnocení

Oprávněný uživatel má při definici šablony možnost výběru vyhodnocovací metody, která z pohledu aplikace specifikuje proces, jakým způsobem bude otázka ohodnocena procentuálně. Změna vyhodnocovací metody ovlivní u této konkrétní aplikace pouze otázky s více možnými odpověďmi, u dalších typů otázek je předpokládáno pouze standardní chování. V systému jsou implementovány dvě vyhodnocovací metody.

- Základní:

Procentuální výsledek závisí pouze na celkovém počtu otázek (Rovnice: 6.1).

$$\frac{S - n}{S + N} * 100$$

S – počet správných odpovědí

N – počet špatných odpovědí

s – počet správně zvolených odpovědí

n – počet špatně zvolených odpovědí

Rovnice: 6.1

- Poměrová:

Procentuální výsledek závisí na poměru správných a špatných odpovědí (Rovnice: 6.2)

$$\frac{S}{S} * 100 - \frac{n}{N} * 100$$

S – počet správných odpovědí

N – počet špatných odpovědí

s – počet správně zvolených odpovědí

n – počet špatně zvolených odpovědí

Rovnice: 6.2

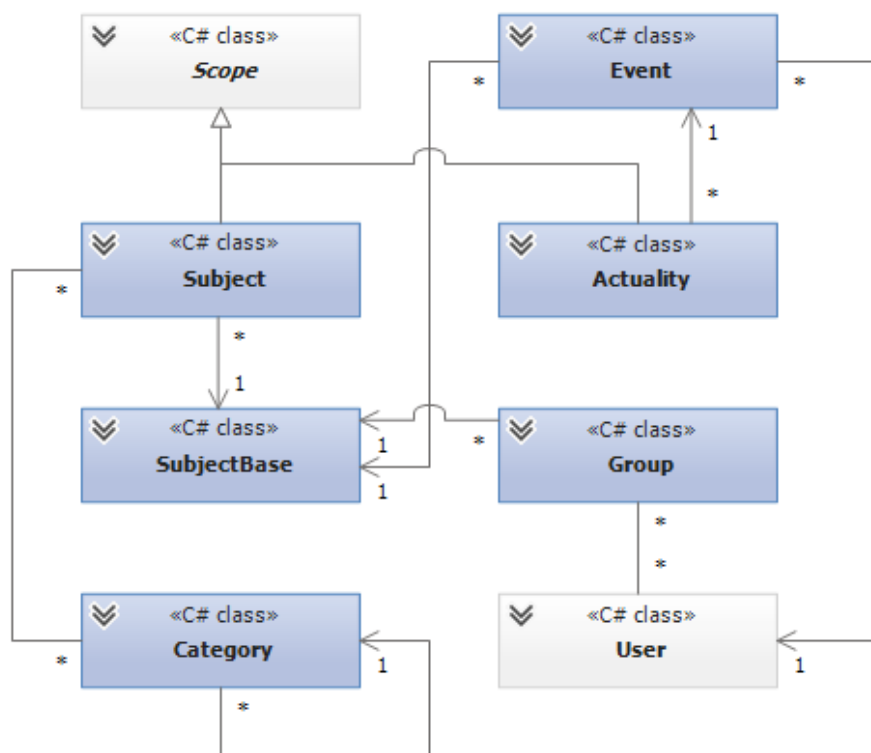
Chování vyhodnocovací metody je možno ovlivňovat dalšími parametry. Lze specifikovat, zda se otázka má vyhodnocovat procentuálně, tedy vypočítávat procentuální úspěšnost, nebo absolutně, kdy se otázka hodnotí pouze sto procenty nebo nula procenty. U procentuálního vyhodnocování se výsledné procenta, v závislosti na nastavení příslušného parametru, mohou pohybovat i do záporných hodnot.

6.3.2 Bodové vyhodnocení

Bodové hodnocení nemůže proběhnout, pokud dříve neproběhlo hodnocení procentuální. Pomocí procentuálního výsledku se poté z maximálního počtu bodů, který má každá otázka definovaný, vypočítá celkový bodový zisk. U tohoto procesu lze určit metodu zaokrouhlení.

6.4 Doménový model

Doménový model je rozdělen do čtyř logických částí. Patří k nim část poskytující informace o dostupných předmětech (Obrázek: 6.2), část realizující autorizační a autentifikační funkčnost (Obrázek: 6.3), část odpovídající za nahrávání souborů (Obrázek: 6.4) a část realizující testování (Obrázek: 6.5). Jednotlivé části jsou mezi sebou propojeny. Na obrázcích doménového modelu jsou bílou barvou odlišeny třídy, které logicky náležejí do jiné části a jsou uvedeny pouze pro znázornění vazeb.



Obrázek: 6.2 – Doménový model – předměty

SubjectBase – Identita předmětu

Subject – Objekt reprezentující data předmětu pro určitý rozsah akademických let

Event – Identita události

Actuality – Objekt reprezentující data aktuality pro určitý rozsah akademických let

Group – Uživatelská skupina

Category – Kategorie náležící předmětu

Material – Objekt reprezentující data studijního materiálu pro určitý rozsah akademických let

Project – Objekt reprezentující data projektu pro určitý rozsah akademických let

ProjectTerm – Termín pro odevzdání určitých úkolů projektu

ProjectFileBase – Identita úkolu projektu

ProjectFile – Objekt reprezentující data úkolu projektu pro určitý rozsah akademických let

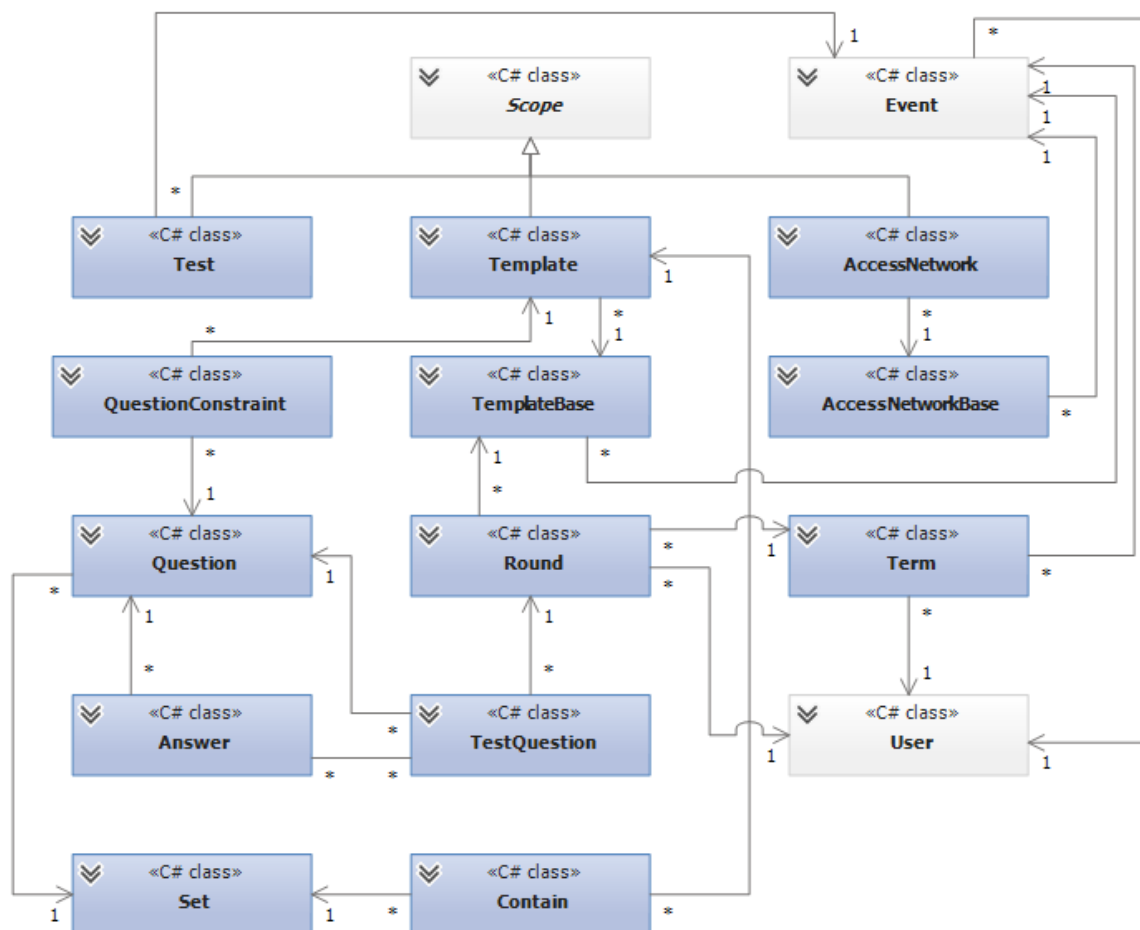
FileFormat – Povolený formát odevzdávaných souborů

Submission – Objekt reprezentující odevzdání úkolu

SubmissionFile – Odevzdaný soubor

Message – Zpráva

Destination – Příjemce zprávy



Obrázek: 6.5 – Doménový model – testování

Test - Objekt reprezentující data testu pro určitý rozsah akademických let

TemplateBase – Identita šablony testu

Template – Objekt reprezentující data šablony testu pro určitý rozsah akademických let, uchovává nastavení konkrétního testu

AccessNetworkBase – Identita sítě, ze které lze test spustit

AccessNetwork – Objekt reprezentující data sítě pro určitý rozsah akademických let

QuestionConstraint – Ovlivnění náhodného výběru otázek jejich vynuceným přidáním nebo odebráním

Question – Objekt reprezentující otázku

Answer – Objekt reprezentující odpověď na otázku

Set – Množina otázek se společnou tematikou

Round – Objekt reprezentující konkrétní pokus o vypracování testu

TestQuestion – Objekt reprezentuje otázku generovanou pro určitý pokus

Term – Termín pro vypracování testu

Contain – Definice obsahu vygenerovaného testu

6.5 Lineární zápis typů databázových entit

Legenda: Primární klíč, *cizí klíč*

AccessNetwork (Id, NetworkAddress, MaskLenght, StartYear, EndYear, *AccessNetworkBaseId*)

AccessNetworkBase (Id, CreatedYear, DeletedYear, *EventId*)

Actuality (Id, Title, Text, StartYear, EndYear, *EventId*)

AllowedAction (Id, Name)

AllowedFormat (*ProjectFile_Id*, *FileFormat_Id*)

Answer (Id, Text, Correct, QuestionId)

Contain (Id, Quantity, Points, *TemplateId*, *SetId*)

Category (Id, ShortName, LongName, *ParentId*)

CategorySubject (*Category_Id*, *Subject_Id*)

Destination (Shown, *MessageId*, *UserLogin*)

Event (Id, AllGroups, CreatedYear, DeletedYear, *Userlogin*, *SubjectBaseId*)

FileFormat (Id, Text)

Group (Id, Name, Key, Year, *SubjectBaseId*)

GroupEvent (*Group_Id*, *Event_Id*)

Logged (*Login*, *Id*)

Material (Id, Name, Description, ContentType, FileName, FileContent, StartYear, EndYear, *EventId*)

Message (Id, Text, SendTime, Year, Public, UserLogin, *ProjectFileBaseId*)

Project (Id, Name, Description, MinPoints, MaxPoints, StartYear, EndYear, *EventId*)

ProjectFile (Id, Name, Description, MinPoints, MaxPoints, StartYear, EndYear, *ProjectFileBaseId*)

ProjectFileBase (Id, CreatedYear, DeletedYear, *EventId*)

ProjectTerm (Id, StartDate, EndDate, Year, *UserLogin*, *EventId*)

ProjectTermProjectFileBase (*ProjectTerm Id*, *ProjectFileBase Id*)

Question (Id, Text, TypeStr, Commentable, Deleted, *SetId*)

QuestionConstraint (*TemplateId*, *QuestionId*, ConstraintTypeStr)

Role (Id, Name, Importance)

RoleAction (*AllowedActionId*, *RoleId*, VerifiGroup, VerifiSubject, VerifiRoleImportance, VerifiOwner)

Round (Id, StartTime, ResumeTime, ExpiredSeconds, Terminated, Locked, *TemplateBaseId*, *TermId*, *RoundNumber*, *UserLogin*)

Set (Id, Name, *SubjectBaseId*)

Subject (Id, Name, Description, StartYear, EndYear, *SubjectBaseId*)

SubjectBase (Id, CreatedYear, DeletedYear)

Submission (*UserId*, *ProjectFileBaseId*, *Year*, StateStr, Points)

SubmissionFile (*UserId*, *ProjectFileBaseId*, *Year*, *Version*, SubmissionDate, ContentType, FileName, Comment, Rating, *ProjectTermId*)

Template (Id, Name, RoundingModeStr, RoundPerQuestion, ValuationMethodStr, ValuationModeStr, NegativeValuation, Locking, UnlimitedLength, TimeLimit, StartYear, EndYear, *TemplateBaseId*)

TemplateBase (Id, CreatedYear, DeletedYear, *EventId*)

Term (Id, Start, End, RoundCount, Year, *EventId*)

Test (Id, Name, Description, MinPoints, MaxPoints, ActiveTemplateId, StartYear, EndYear, *EventId*)

TestQuestion (*RoundId*, *QuestionId*, Comment, TextAnswer, MaxPoints, Percentage, Order)

User (Login, FName, LName, Email, Password, Type, *RoleId*)

UserAnswer (*TestQuestion RoundId*, *TestQuestion QuestionId*, *Answer Id*)

7 Implementace

Kapitola je věnována samotné implementaci navržené aplikace. Jsou zde rozebrány použité technologie a popsáno řešení konkrétních funkčních bloků z hlediska vývoje.

Aplikace byla implementována pomocí technologií ASP.NET 4.0 podle softwarové architektury MVC. Pro přístup k datům a práci s nimi byl použit Entity framework 5.0. Tyto technologie byly podrobněji popsány ve čtvrté kapitole (Webové technologie platformy .NET). Bylo použito vývojové prostředí Visual Studio 2012 Ultimate.

Při vývoji byla aplikace spouštěna na serveru IIS Express s využitím lokálního databázového serveru SQL Express.

7.1 Přístup k datům

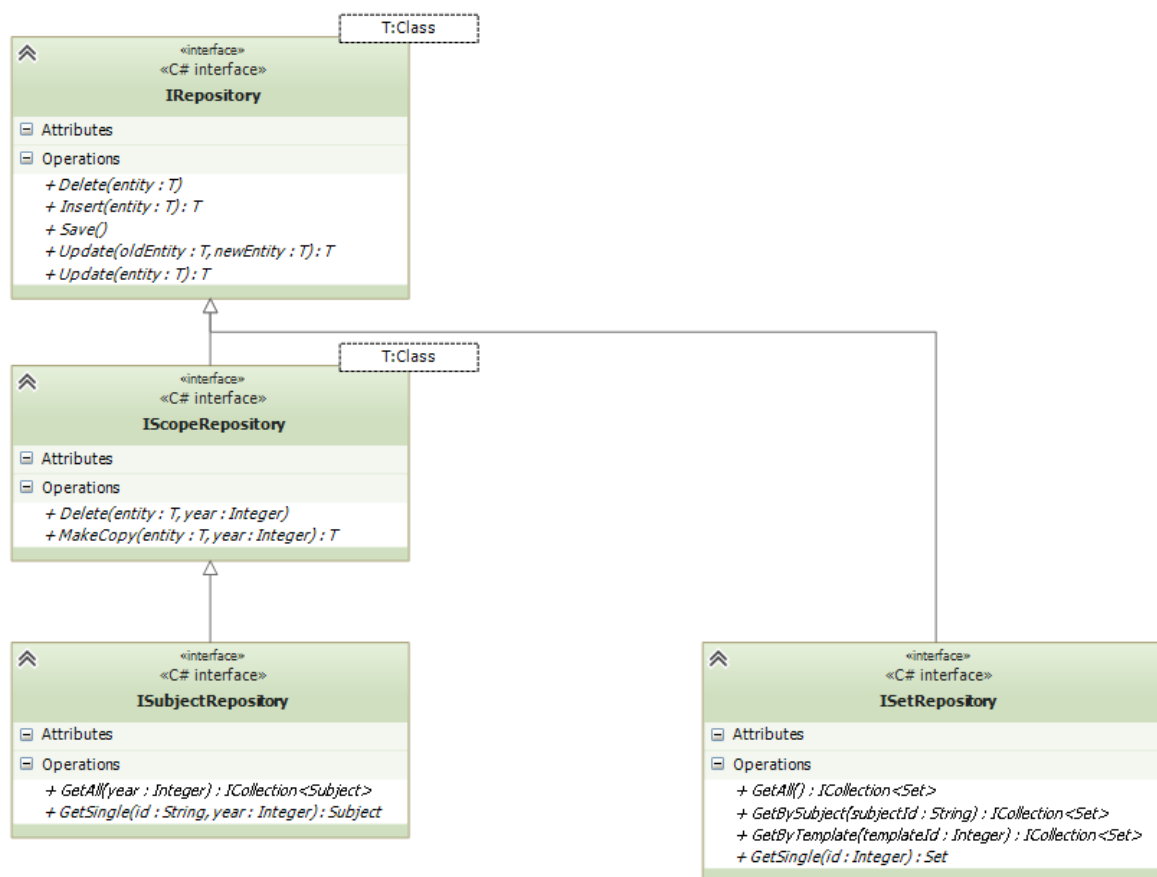
Zajištění přístupu k datům a implementace databázové struktury byla z velké části ponechána v režii Entity Frameworku. Byla použita technika známá jako „Code first“, to znamená, že struktura databáze je řízená podle doménového modelu aplikace. Doménový model byl v předchozí kapitole rozdělen na čtyři logické části realizující samostatnou funkčnost. Z pohledu implementace je každá část členěna do vlastního jmenného prostoru:

1. Část zodpovědná za realizaci online testování studentů (`Data.Domain.Testing`)
2. Část realizující autentifikační a autorizační funkčnost (`Data.Domain.Security`)
3. Část poskytující informace o dostupných předmětech (`Data.Domain.Subjects`)
4. Část pro odevzdávání a kontrolu souborů (`Data.Domain.Uploading`)

Jednotlivé doménové třídy přesně odpovídají tabulkám v databázi s tím, že pro mapování vazeb cizích klíčů jsou přidány navigační vlastnosti, které udržují reference na příslušné objekty.

Vrstva přístupu k datům je implementována pomocí návrhového vzoru Repository [11], který se stará o získání, úpravu, mazání a vkládání dat. Výhodou tohoto řešení je fakt, že o práci s daty se starají cizí objekty vzájemně nezávislé, tudíž v případě nutnosti je lze, díky implementaci rozhraní, snadno nahradit jinými (například při přechodu na jinou databázi).

V této konkrétní aplikaci existuje generické rozhraní, které obsahuje metody `Insert`, `Update`, `Delete` a `Save`. Chování těchto metod je stejné pro všechny doménové objekty. Pro specifitější požadavky je toto rozhraní patřičně rozšířeno. Obrázek: 7.1 znázorňuje zmíněné generické rozhraní `IRepository` a rozšiřující rozhraní `ISetRepository`, které obsahuje specifické metody pro selekci otázkových setů (`GetAll`, `GetBySubject`, `GetByTemplate`, `GetSingle`). Rozhraní `IScopeRepository` poskytuje potřebné funkce pro práci s objekty, které jsou dostupné jen v určitém období akademických let. Bude podrobněji popsáno v kapitole Uchovávání historie předmětů a událostí.

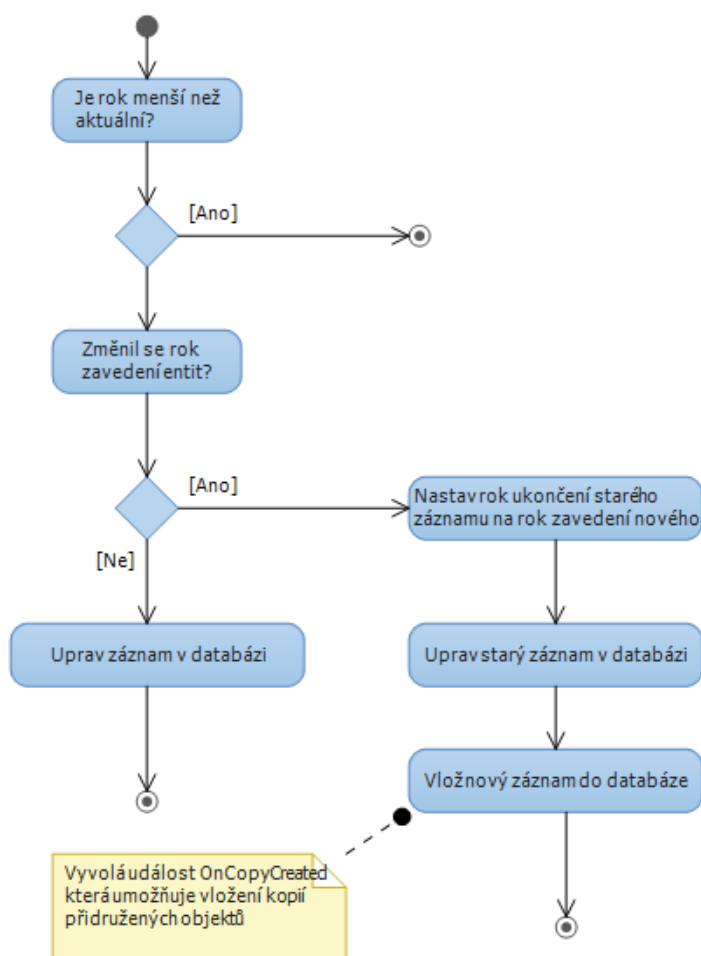


Obrázek: 7.1 – Příklad implementace vrstvy přístupu k datům

7.2 Uchovávání historie předmětů a událostí

Informace o předmětu a událostech, jejich nastavení a podmínky se mohou v průběhu času měnit. Aplikace je navržena tak, aby uchovávala změny pouze se změnou akademického roku, to znamená, že změny v tomtéž akademickém roce nebudou ukládány a staré údaje budou ztraceny.

Doménové objekty, které jsou schopny uchovávat změny, rozšiřují abstraktní třídu *Scope*. Ta obsahuje vlastnosti indikující rok zavedení a rok ukončení (*StartYear*, *EndYear*), kdy pro konkrétní akademický rok může existovat jen jedna instance. Rozhraní *IScopeRepository* (Obrázek: 7.1) omezuje generický typ na typ rozšiřující třídu *Scope*. Třída *ScopeRepository* implementující toto rozhraní, přepisuje metodu *Update*, tak aby se staré údaje zachovaly a nové přidaly (Obrázek: 7.2).



Obrázek: 7.2 – Diagram aktivit – úprava záznamu

Toto chování se předpokládá i pro mazání záznamu. Rozhraní `IScopeRepository` přetěžuje metodu `Delete` a přidává parametr `Year`, který indikuje akademický rok smazání záznamu. Data se odstraňují z databáze pouze v případě, že byly vytvořeny v tomtéž akademickém roce, v opačném případě se pouze označí jako smazané (`EndYear` se nastaví na příslušný akademický rok).

7.3 Zabezpečení

7.3.1 Autentifikace

Pro ověření identity uživatele se primárně ověřují vstupní údaje vůči databázi. Při nenalezení záznamu může být prohledán vzdálený server s ActiveDirectory pomocí LDAP protokolu [9], kdy je záznam při kladném výsledku zkopírován do databáze.

Externím uživatelům je umožněna registrace, a je jim generován login s ohledem na univerzitní záznamy.

7.3.2 Autorizace

Určení privilegií spočívá v rozhodnutí, zda je uživatel oprávněn vykonávat danou akci. Množina dostupných akcí, je jasně definována a uložena jako tabulka v databázi. Pro zajištění jistého stupně nezávislosti je mezi akce a uživatele přidána vrstva rolí. Role mohou být generovány dynamicky a nejsou nijak závislé na aplikaci, ovšem v tomto konkrétním případě jsou role předem známe a není zde tedy patřičně využita výhoda jejich nezávislosti.

V některých případech pouhé ověření vůči akci nestačí a je třeba ověřit další záležitosti. Například konkrétnímu uživateli, který je v roli student by měl být umožněn přístup pouze ke svým odevzdaným dokumentům, pokud by se pokusil nahlédnout do projektů jiných studentů (např. přepsáním URL adresy), aplikace mu v tom musí zabránit. Naproti tomu uživatel v roli instruktor, má přístup ke všem dokumentům, s výjimkou dalších omezení (například jen dokumenty v rámci jeho skupiny).

Aplikace je navržena tak aby ověřovala následující možnosti:

- Ověření zda je uživatel vlastník daného objektu
- Ověření zda má role větší oprávnění než jiná
- Ověření zda je uživatel členem dané skupiny
- Ověření zda je uživatel přihlášen na předmět

7.4 Testování

7.4.1 Generování testů

Generování testových otázek zajišťuje třída `Generator`. Pro generování je nutné nastavit aktivní šablonu testu. Postupným procházením kolekce otázkových setů v šabloně, se provede náhodný výběr otázek z databáze. Aby nedocházelo k duplicitě výběru, jsou již vygenerované otázky přidány do pomocné kolekce ignorovaných otázek, která je při každém dotazu zohledněna. Generování otázek probíhá ve třech krocích:

1. Přidání odstraněných otázek do kolekce ignorovaných otázek
2. Přidání přiřazených otázek do výsledku a zároveň do kolekce ignorovaných otázek
3. Doplnění výsledku náhodným výběrem, a přidání výsledku do kolekce ignorovaných otázek

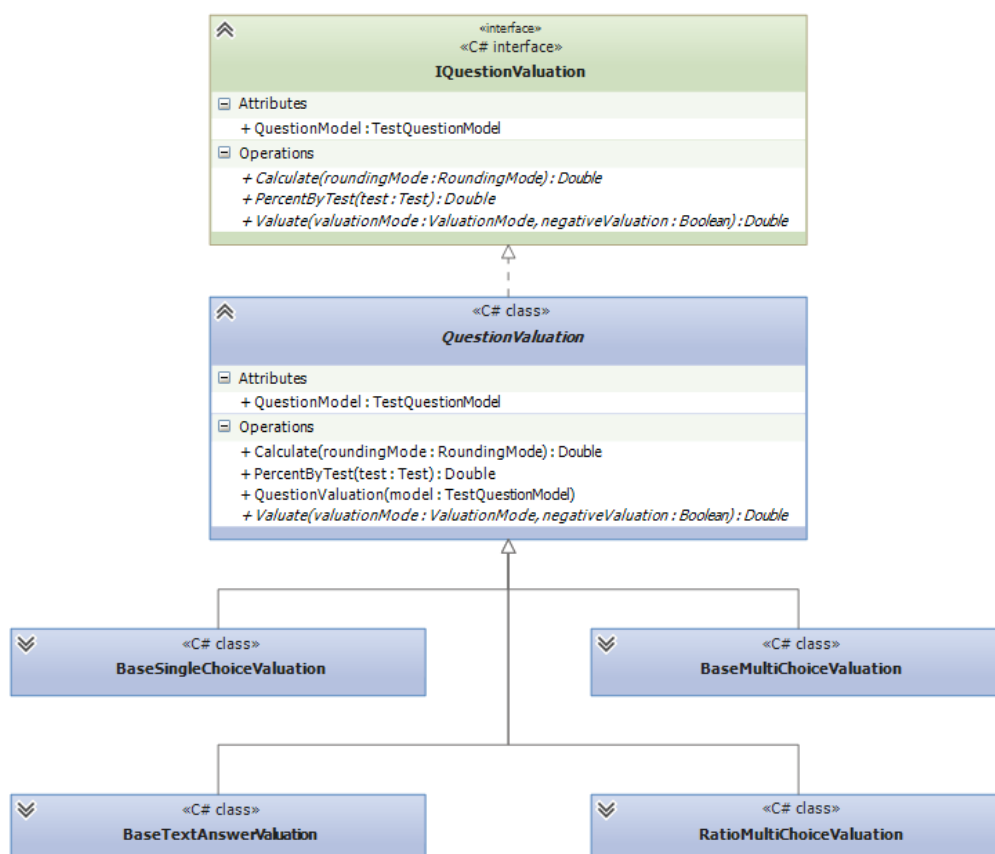
7.4.2 Vyhodnocování testů

Testové otázky jsou vyhodnoceny hromadně až po ukončení testu uživatelem, případně při vypršení časového limitu. Pro každou otázku je vytvořen vlastní objekt, který ji umí vyhodnotit jak procentuálně tak bodově. K vytvoření těchto objektů je použit vytvářecí návrhový vzor `Abstract Factory` [10]. Tvorba konkrétní továrny závisí na vyhodnocovací metodě zvolené v aktivní šabloně testu. Tyto metody byly popsány v předchozí kapitole a jsou reprezentovány třídami `BaseValuationFactory` a `RatioValuationFactory`. Tvorba konkrétního objektu, který je použit pro vyhodnocení otázky, závisí na typu příslušné otázky (Tabulka: 7.1).

	BaseValuationFactory	RatioValuationFactory
Otázka s jednou možnou odpovědí	BaseSingleChoiceValuation	BaseSingleChoiceValuation
Otázka s více možnými odpověďmi	BaseMultiChoiceValuation	RatioMultiChoiceValuation
Otázka s textovou odpovědí	BaseTextAnswerValuation	BaseTextAnswerValuation

Tabulka: 7.1 – Přehled vytvářených objektů pro hodnocení otázek testů

Použití vzoru Abstract Factory se v tomto případě může jevit jako zbytečné, jelikož změny v chování se projeví jen u otázky s více možnými odpověďmi, na druhou stranu dovoluje snadnou rozšiřitelnost a pro přidání další typů otázek stačí pouze implementovat rozhraní `IQuestionValuation` (Obrázek: 7.3). Procentuální výsledek konkrétní otázky vrací metoda `Valuate`, zatímco metoda `Calculate` vrací bodový zisk.



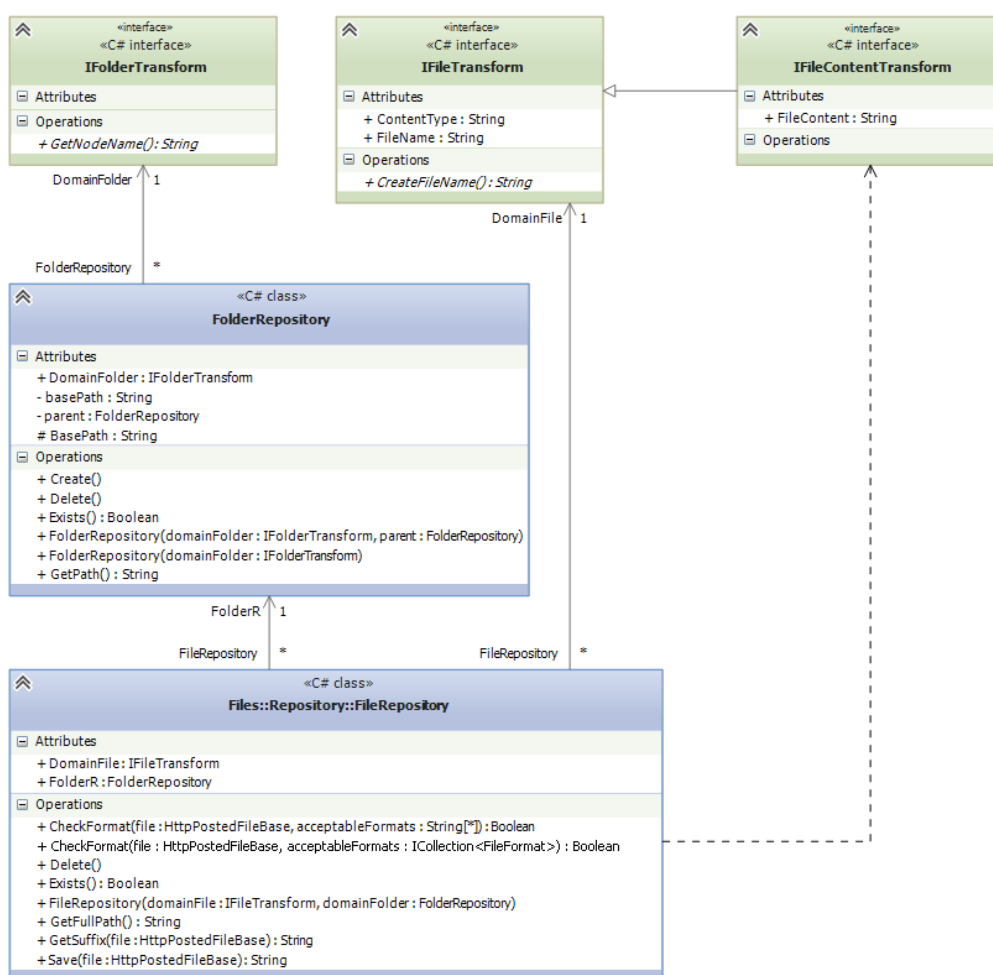
Obrázek: 7.3 – Třídní diagram - Vyhodnocení testových otázek

7.5 Odevzdávání projektů a nahrávání studijních materiálů

7.5.1 Adresářová struktura pro ukládání souborů

Po vytvoření příslušného předmětu je v adresáři files, nacházející se v datovém úložišti aplikace (adresář App_Data), vytvořen podadresář s názvem stejným jako Id předmětu. Pro správné fungování aplikace je nutné správně nastavit práva zápisu. Pro soubory odevzdávané v rámci projektu je struktura dále větvena a pro každý projekt je generován vlastní adresář.

Z pohledu implementace byly pro práci se soubory přidány rozhraní `IFileTransform`, `IFileContentTransform` a `IFolderTransform`. Rozhraní definují metody potřebné k mapování databázových entit na soubory nebo složky v adresářové struktuře. Doménové objekty, které implementují tato rozhraní, pak mohou být zpracovávány pomocí reposiářů `FileRepository` nebo `FolderRepository`.



Obrázek: 7.4 – Třidní diagram – práce se soubory

Metoda `GetNodeName()` rozhraní `IFolderTransform` vrací název příslušného adresáře. V aplikaci toto rozhraní implementují doménové třídy `Event`, reprezentující událost, a `Subject`, reprezentující předmět (viz: Obrázek: 6.2 – Doménový model – předměty). V obou případech metoda vrací pouze `Id`.

Metoda `CreateFileName()` rozhraní `IFileTransform` vrací název, pod jakým bude soubor uložen v adresářové struktuře. V aplikaci toto rozhraní implementuje doménová třída `SubmissionFile`, která reprezentuje konkrétní odevzdaný soubor. Metoda, v tomto případě vrací řetězec v následujícím formátu: `id úkolu-login uživatele-verze-rok`. (Viz: Obrázek: 6.4 – Doménový model – nahrávání souborů).

Doménová třída `Material`, reprezentující studijní materiál, implementuje rozšiřující rozhraní `IFileContentTransform`, které navíc umožňuje uložit obsah souboru do databáze pro účely vyhledávání. Čtení obsahu zajišťují třídy dědicí od abstrakce `ContentReader`, kdy každá má přiděleny jiné formáty vstupních souborů. Za tímto účelem byly do aplikace přidány balíčky `iTextSharp`, pro práci se soubory pdf, `Spire.Doc`, podporující formáty doc a docx, a `OpenXml 2.0`, pro čtení pptx prezentací. Podporovány jsou i textové soubory txt. Název studijních materiálů, pod kterým jsou soubory ukládány do adresářové struktury, tvoří pouze `Id`.

7.5.2 Stav y odevzdání

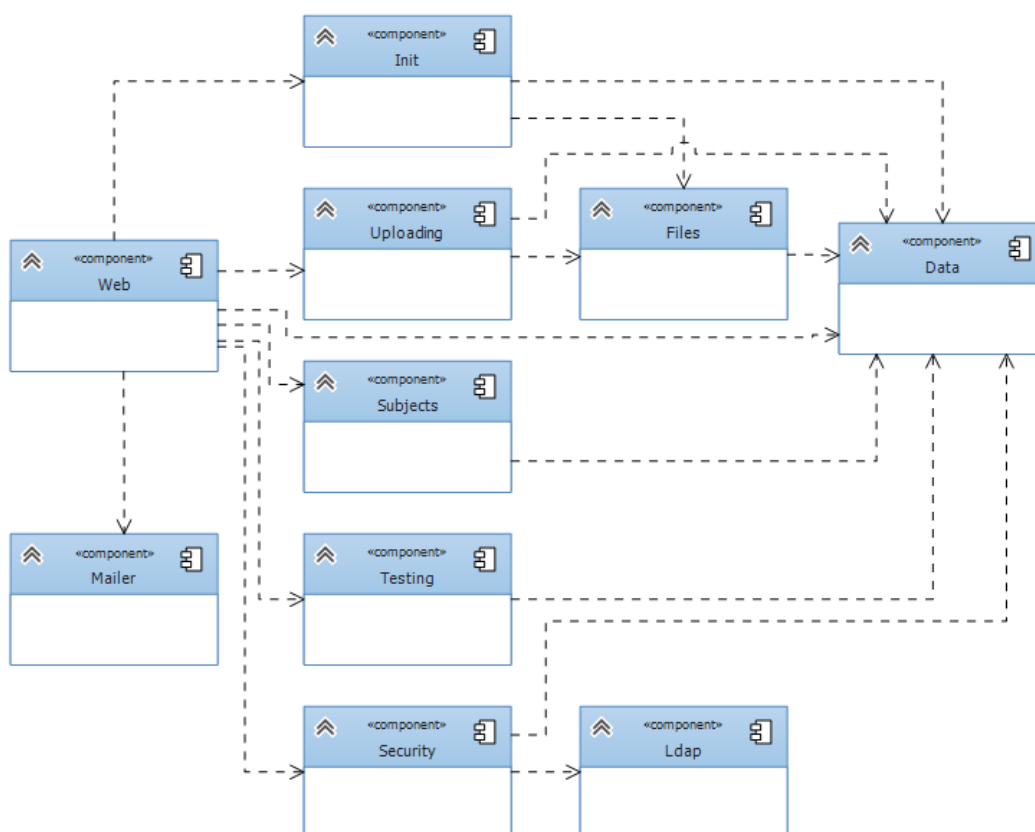
Stav konkrétních projektových úkolů pro konkrétní studenty je uložen v objektu doménové třídy `Submission` ve vlastnosti `State`. Aplikace rozeznává následující stavy:

- Neodevzdáno:
Zatím nebyl odevzdán žádný soubor
- Odevzdáno:
Úkol je odevzdán a čeká na zkontrolování. Pokud uživatel nahraje další soubor, je ten stávající přepsán.
- K přepracování:
Úkol je třeba dopracovat a nahrát znovu. Nový soubor je uložen jako nová verze. Stávající soubory jsou zachovány. Po nahrání dalšího souboru je stav změněn na „Odevzdáno“.
- Zamítnuto:
Úkol je zamítnut, je hodnocen nedostačujícím počtem bodů.
- Schváleno:
Úkol schválen, je hodnocen dostačujícím počtem bodů

7.6 Diagram komponent

- Web:
Vstupní bod celé aplikace. Jde o webové rozhraní, díky kterému mohou uživatelé s aplikací pracovat. Obsahuje pouze obsluhu vstupních požadavků a ověření uživatelem zadaných dat, eventuálně základní provádění CRUD operací. Logika aplikace je ponechána na jiných komponentách.
- Init:
Slouží pro realizaci jakýchkoliv inicializačních funkcí. Při vývoji byla komponenta primárně používána pro inicializaci databáze testovacími daty.
- Uploading:
Komponenta obstarávající logiku odevzdávání projektů a nahrávání studijních materiálů.
- Subjects:
Komponenta zajišťující správu předmětů a studijních skupin.

- Security:
Komponenta realizující autorizační a autentifikační funkčnost.
- Files:
Zajišťuje práci se souborovým systémem. Ukládá soubory a spravuje adresáře pro potřeby aplikace.
- Global:
Komponenta poskytující třídy přístupné ze všech ostatních komponent. Obsahuje převážně výjimky a rozšiřující metody (Helpery).
- Testing:
Komponenta obstarávající logiku spravování a generování testů.
- Data:
Komponenta obsahuje doménový model aplikace a zajišťuje přístup k datům.
- Mailer:
Komponenta potřebná pro odesílání emailů.
- Ldap:
Pokud je nastaveno ověřování vůči serveru pomocí LDAP protokolu (Web.config), komponenta se připojí k určenému serveru a na základě vstupních informací ověří identitu uživatele.



Obrázek: 7.5 – Diagram komponent

8 Aplikace

8.1 Uživatelské rozhraní

Vstupním bodem aplikace je stránka s výpisem jednotlivých předmětů, dostupných v daném akademickém roce, které jsou členěny do kategorií. V předmětech lze vyhledávat, a to dle jejich názvu, popisu a příslušných studijních materiálů (viz: 7.5.1). Po vstupu do detailu předmětu, se v levé části zobrazí navigační menu obsahující akce, které jsou uživateli k dispozici, a které může v závislosti na autorizačním a autentifikačním systému aplikace vykonávat. Menu mimo jiné naviguje na seznamy událostí (test, projekt, studijní materiál, aktualita), které jsou v rámci předmětu a povolených skupin k dispozici. Po kliknutí na některou událost se navigační menu rozšíří.

8.1.1 Navigační menu předmětu

Jak již bylo zmíněno, menu ke konkrétnímu předmětu obsahuje volbu seznamu událostí, která je dostupná všem přihlášeným uživatelům. Pro oprávněné uživatele jsou k dispozici volby „Skupiny“ a „Správce setů“.

Volba „Správce setů“ umožňuje uživateli spravovat otázkové sety a konkrétní otázky v nich (Obrázek: 8.1). Mimo jiné obsahuje i procentuální úspěšnosti jednotlivých otázek. Jelikož je možno sety využívat na více testech, je volba řazena zde a ne v navigačním menu testu.

The screenshot shows the 'Repositář studijních materiálů' application. The top header is green with the university logo and user information: 'ber0049 [Změnit heslo, Odhlásit se]' and 'IP: 89.103.68.57'. Below the header, there are tabs for 'Předměty', 'Uživatelé', and 'Nápověda'. The main content area is titled 'Fyzika' and 'Seznam otázek'. On the left, there is a sidebar with icons and labels: 'Fyz', 'Informace', 'Studijní materiály', 'Projekty', 'Testy', 'Skupiny', and 'Správce setů'. The main area shows a table of question sets. The table has columns: 'Text', 'Typ', 'Set', 'Odpovědi', 'Použito', and 'Úspěšnost'. The data rows are:

Text	Typ	Set	Odpovědi	Použito	Úspěšnost
Napětí a proud jsou ve fázi:	Jedna odpověď	Střídavý proud	1 / 3	2	50
Amplituda střídavého napětí Um ...	Jedna odpověď	Střídavý proud	1 / 3	4	50
V trojfázové rozvodné síti je v ...	Jedna odpověď	Střídavý proud	1 / 3	5	0

Below the table is a 'Přidat otázku' button. On the right, there is a 'Kontakt' form with fields for 'Předmět' and 'Kontaktní email', and an 'Odeslat' button. The footer shows '©2013 Repository, Html, Css, ASP.NET - by Marek Berger'.

Obrázek: 8.1 – Správce otázkových setů

8.1.2 Navigační menu testu

Obsah navigačního menu se opět liší v závislosti na přihlášeném uživateli. Studentovi je k dispozici volba „Vykonání“, která ho přesune na stránku zobrazující dostupné termíny, kdy může test vykonat, a seznam pokusů, které již absolvoval. Jeho výsledky jsou mu tedy zpětně k dispozici (Obrázek: 8.2).

Repositář studijních materiálů

skc0001 [Změnit heslo, Odhlásit se]
IP: 89.103.68.57

Předměty Uživatelé Nápověda

Testovací - pokus: 1

► Celkový výsledek: 40% / 40 ✓

0 %
Body: 0 / 20

Pohybuje-li se elektron v elektrickém poli působením pouze elektrostatických sil, nemění se jeho:

- ☐ kinetická energie
- ☒ potencionální energie
- ☐ hybnost
- ☐ celková energie

0 %
Body: 0 / 20

Zdrojem stacionárního magnetického pole je:

- ☐ Pohybující se vodič s konstantním proudem
- ☒ nepohybující se vodič s časově proměnným proudem
- ☐ pohybující se permanentní magnet
- ☐ nepohybující se vodič s konstantním proudem

100 %
Body: 20 / 20

Akademický rok
2012/2013

Kontakt

Předmět

Kontaktní email

Odeslat

Obrázek: 8.2 – Výsledek testu

Ostatním rolím se naopak zobrazí volba „Výsledky“, kde je vyobrazené nejvyšší dosažené hodnocení pro uživatele, přihlášené do příslušných skupin (Obrázek: 8.5). Stránka umožňuje zobrazení všech pokusů jednotlivě pro konkrétního uživatele. Pokud je test probíhající, je možné ho z této stránky uzamknout nebo naopak odemknout.

Dále se pro oprávněné uživatele zobrazí volby „Přístupné sítě“, umožňující spravovat sítě, ze kterých lze test spustit, „Termíny“ umožňující přidávat termín k vykonání testu pro všechny nebo konkrétního studenta a „Šablony“ obsahující komplexní funkčnost pro řízení testů.

Volba „Šablony“ zobrazí stránku s možností výběru aktivní šablony, která bude při následujícím vykonání testu použita. Výběr lze realizovat pouze u šablon, které mají správně rozvržené management otázek, tedy jejich celkové bodové hodnocení je ekvivalentní s maximální bodovým hodnocením testu. Příklad správného rozvržení zobrazuje Obrázek: 8.3.

Fyzika

Fyz

- Informace
- Studijní materiály
- Projekty
- Testy
- Skupiny
- Správce setů

Test

- Informace
- Šablony
- Výsledky
- Termíny
- Přístupné sítě

Pětiminutovka

Rozděleno: 100 / 100

Set	Počet otázek	Body	
Elektrický náboj a elektrické pole	1	20	Smazat
Elektrický proud	1	20	Smazat
Elektrický proud v polovodičích	1	20	Smazat
Elektrický proud v plynech	1	20	Smazat
Střídavý proud	1	20	Smazat
Elektrický proud v plynech	0 (4)	0	Uložit

Elektrický proud v plynech

Elektrický proud v plynech

Text	Typ	Set		
Proces, při kterém se plyn stává vodivým, se nazývá	Jedna odpověď	Elektrický proud v plynech	-	
V plynech se uplatňuje:	Jedna odpověď	Elektrický proud v plynech	-	
Plyn lze ionizovat:	Více odpovědí	Elektrický proud v plynech	-	
Přenos náboje v plynu se může uskutečňovat:	Jedna odpověď	Elektrický proud v plynech	-	
Katodové záření	Více odpovědí	Elektrický proud v plynech	-	

[Zpět](#)

Obrázek: 8.3 – Rozvržení otázek v šabloně

8.1.3 Navigační menu projektu

Toto navigační menu se zobrazí po výběru konkrétního projektu ze seznamu. Všichni přihlášení uživatelé mají k dispozici volbu „Úkoly“, což je z pohledu studenta prostředí pro odevzdání požadovaných souborů a diskuzi se cvičícími, zatímco ostatní role mohou, v závislosti na oprávnění, úkoly spravovat.

Ke kontrole projektu se oprávnění uživatelé dostanou pomocí volby Hodnocení, kde je v přehledu zobrazen stav odevzdání zvoleného úkolu uživatelů v příslušných skupinách (Obrázek: 8.4). Tento seznam zobrazuje pouze konkrétní úkoly. Pro zobrazení celkových výsledků slouží volba „Výsledky“.

Fyzika

▼ Fyz

Hodnocení

První úkol

Login	Jméno	Příjmení	Skupina	Stav	Body	Akce
chw0010	Xenie	Chwalková	Testovací	Neodevzdáno	-	(0)
skc0001	Jiří	Šček	Testovací	Odevzdáno	-	(0)

▼ Projekt

Informace
Úkoly
Termíny
Hodnocení
Výsledky

Obrázek: 8.4 – Hodnocení projektu

8.2 Možnosti rozšíření

8.2.1 Videokonference a možnost sdílení plochy

Toto rozšíření by bylo cíleno zejména na kombinovanou formu studia. Konference by se realizovala v rámci studijní skupiny. K vytvoření konference by měl právo pouze cvičící, případně administrátor.

8.2.2 Graf zobrazující statistiky úspěšnosti otázek

Pokud budou studenti u testu dosahovat podezřele vysokého hodnocení, je možné, že jsou otázky již mezi studenty rozšířené a známé. Takové otázky je třeba u dalších testů vynechat a nahradit je novými. Právě graf, zobrazující procentuální úspěšnost konkrétní otázky v závislosti na čase, by pomohl takové otázky odhalovat. Graf by mohl zobrazovat i procentuální podíly volby konkrétních odpovědí.

8.2.3 Přenositelnost studijních materiálů

Aplikace nepodléhá normě SCORM a není tak uzpůsobená pro práci s externím výukovým obsahem. Kdyby se toto změnilo, mělo by to přínos pro kvalitu studijních materiálů i celkovou kvalitu výuky. Studijní materiály by mohly být importovány z jiných již existujících LMS.

8.2.4 Další vyhodnocovací metody

Tyto metody zásadně ovlivňují procentuální zisk jednotlivých testů. Aplikace implementuje jen dvě metody (kapitola: Procentuální vyhodnocení 6.3.1) a je navržena tak, aby bylo snadné implementovat další (kapitola: 7.4.2).

Případně by se aplikace mohla rozšířit o uživatelský vstup, kde by oprávněný uživatel mohl zadat vlastní vzorec vyhodnocovací metody. Tento vstup by se musel patřičně validovat.

8.2.5 Implementace webové služby

Implementace třívrstvé architektury by umožnila použití aplikace, případně jen určitých částí, i na jiných platformách. Příkladem účelného využití je implementace samostatné Windows aplikace, představující prostředí pro realizaci testování. Výhodou by byly širší možnosti zabezpečení a podobně.

8.3 Testování

Pro účely testování byla aplikace nahrána na univerzitní server a spouštěna pomocí webového serveru Microsoft IIS 7. Roli databázového serveru představoval Microsoft SQL Server 2008 R2. Aplikace byla přístupná z URL <http://scal.vsb.cz>.

Testování probíhalo ve spolupráci s gymnáziem Ostrava – Hrabůvka (Františka Hajdy 34). Celkem 11 studentů třetího ročníku se testování zúčastnilo a vše proběhlo bez větších problémů. Testováno bylo jen prostředí pro generování testů. Pro tyto účely bylo nahráno přes třicet otázek. Výsledky testů znázorňuje Obrázek: 8.5.

Fyzika								
Výsledky								
Login	Jméno	Příjmení	Skupina	Procenta	Body	Splněno		
ara0007	Kraft	Araf	G3	80	80	✓		
duo0001	David	Duong	G3	80	80	✓		
hil0024	Vaxi	Hill	G3	20	20	✗		
ko.0001	Luc	Ko.	G3	80	80	✓		
mil0042	Šimona	Milotová	G3	20	20	✗		
pop0042	Petr	Popule	G3	20	20	✗		
sep0001	Eva	Šeptáková	G3	36,67	37	✗		
spa0091	spartakus	spartakus	G3	60	60	✓		
uhr0020	Katerina	Uhrova	G3	20	20	✗		
ver0028	Jaroslav	Verlík	G3	60	60	✓		
vit0077	Lucie	Vitoslavská	G3	26,67	27	✗		

Obrázek: 8.5 – Výsledky testů

9 Závěr

Cílem práce bylo navrhnout, implementovat a otestovat nástroj pro hromadné testování studentů, odevzdávání projektů a publikování studijních materiálů. Toto zadání částečně spadá do definice LMS systémů, které byly, spolu s problematikou e – learningu, obsahem druhé kapitoly. Při návrhu aplikace byly zohledněny stávající systémy. Design a uživatelské rozhraní aplikace byly navrženy tak, aby pro uživatele byly jednoduché, přehledné a snadno pochopitelné. Z hlediska funkcionality bylo do jisté míry přihlédnuto k existujícím systémům, které byly popsány ve třetí kapitole. Vlastní vývoj aplikace, od specifikace požadavků přes návrh až po implementaci a testování byl rozebírán od čtvrté až po osmou kapitolu.

Implementaci aplikace doprovázelo několik problémů, z nichž některé byly v práci popsány detailněji. K řešení bylo ve většině případů užito návrhových vzorů případně jiných ověřených postupů, což mimo jiné usnadňuje budoucí rozšíření.

Návrhy na další rozšíření, zmíněné v sedmé kapitole, by pro aplikaci byly přínosem z hlediska reálného používání v praxi. Ovšem dle mého názoru má nejen tato aplikace či jiné LMS systémy ale i e – learning jako takový obrovský potenciál pro další vývoj a zdokonalování.

Použitá literatura

- [1] E-learning - nástroje pro tvorbu a řízení výuky. *Volny* [online]. 2007 [cit. 2013-04-19]. Dostupné z: http://www.volny.cz/xmichalx/bp/xnovm133_BP.htm#_Toc191661312
- [2] STRÍTESKÁ, Hana. Historie e-learningu v České republice. *Fakulta informatiky Masarykovy univerzity* [online]. 2003 [cit. 2013-04-19]. Dostupné z: <http://www.fi.muni.cz/usr/jkucera/pv109/2003p/xstrites.htm>
- [3] Virtuální univerzity. OBSTOVÁ, Gabriela. *Inflow* [online]. 2010 [cit. 2013-04-19]. Dostupné z: <http://www.inflow.cz/virtualni-univerzity>
- [4] LCMS a LMS, vývoj kurzů. PEJŠA, Jan. *Kontis elearning* [online]. [cit. 2013-04-19]. Dostupné z: http://www.e-learn.cz/soubory/LMS_LCMS.pdf
- [5] Moodle. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001-2013, 7. 3. 2013 [cit. 2013-03-23]. Dostupné z: <http://cs.wikipedia.org/wiki/Moodle>
- [6] iTutor. *Kontis elearning* [online]. 2003 [cit. 2013-04-23]. Dostupné z: <http://www.kontis.cz/soubory/iTutor.pdf>
- [7] BUDAI, David. Oracle: Příběh společnosti, která popadla příležitost a nezklamala. MAREK, Vlastimil. *Itbiz* [online]. 2010 [cit. 2013-04-19]. Dostupné z: <http://www.itbiz.cz/oracle-pribeh-spolecnosti-ktera-popadla-prilezitost-nezklamala>
- [8] FREEMAN, Adam a Steven SANDERSON. *Pro Asp.net MVC 3 framework*. 3rd ed. New York: Apress, 2011. ISBN 978-143-0234-050.
- [9] ROBINSON, Simon. *C#: programujeme profesionálně*. 1. vyd. Brno: Computer Press, 2003, xxx, 1130 s. ISBN 80-251-0085-5.
- [10] BISHOP, Judith. *C# - Návrhové vzory*. Brno: ZONER software, a.s., 2010. ISBN 978-80-7413-076-2.
- [11] DYKSTRA, Tom. Implementing the Repository and Unit of Work Patterns in an ASP.NET MVC Application. *ASP.NET* [online]. 2011 [cit. 2013-04-19]. Dostupné z: <http://www.asp.net/mvc/tutorials/getting-started-with-ef-using-mvc/implementing-the-repository-and-unit-of-work-patterns-in-an-asp-net-mvc-application>

Seznam obrázků

Obrázek: 3.1 – LMS Moodle – tvorba otázky.....	5
Obrázek: 3.2 – iTutor tester – tvorba otázky.....	6
Obrázek: 3.3 – Oracle Academy – vypracování testu.....	7
Obrázek: 3.4 – Cisco Networking Academy - studijní materiály.....	7
Obrázek: 3.5 – Microsoft Virtual Academy.....	8
Obrázek: 4.1 – MVC architektura	10
Obrázek: 6.1 – Diagram případu užití.....	18
Obrázek: 6.2 – Doménový model – předměty.....	20
Obrázek: 6.3 – Doménový model – autorizace a autentifikace	21
Obrázek: 6.4 – Doménový model – nahrávání souborů.....	21
Obrázek: 6.5 – Doménový model – testování	22
Obrázek: 7.1 – Příklad implementace vrstvy přístupu k datům.....	26
Obrázek: 7.2 – Diagram aktivit – úprava záznamu	27
Obrázek: 7.3 – Třídní diagram - Vyhodnocení testových otázek.....	29
Obrázek: 7.4 – Třídní diagram – práce se soubory.....	30
Obrázek: 7.5 – Diagram komponent.....	32
Obrázek: 8.1 – Správce otázkových setů.....	33
Obrázek: 8.2 – Výsledek testu	34
Obrázek: 8.3 – Rozvržení otázek v šabloně.....	35
Obrázek: 8.4 – Hodnocení projektu.....	36
Obrázek: 8.5 – Výsledky testů	37

Seznam rovnic

Rovnice: 6.1.....	19
Rovnice: 6.2.....	19

Seznam tabulek

Tabulka: 7.1 – Přehled vytvářených objektů pro hodnocení otázek testů.....	29
--	----

Seznam příloh

<i>Příloha: A - Programátorská příručka</i>	<i>I</i>
---	----------

Součástí bakalářské práce je CD.

Adresářová struktura přiloženého CD:

- Source – Zdrojové kódy aplikace
- Publish – Publikovaná aplikace